Risto Laakso

Algorithmic description of pyramidal neuron populations from mouse neocortex

Faculty of Electronics, Communications and Automation

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 29.10.2010

Thesis supervisor:

Professor Jouko Lampinen

Thesis instructor:

Assistant Prof. Patrik Krieger

Aalto University
School of Science
and Technology

AALTO-YLIOPISTO
TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Risto Laakso

Työn nimi: Algoritminen kuvaus hiiren etuaivojen pyramiidisista neuronipopulaatioista

Päivämäärä: 29.10.2010 Kieli: Englanti Sivumäärä:6+61

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Lääketieteellisen tekniikan ja laskennallisen tieteen laitos

Professuuri: Laskennallinen tekniikka Koodi: S-114

Valvoja: Professor Jouko Lampinen

Ohjaaja: Assistant Prof. Patrik Krieger

Groh et al (2010) tutkivat suuria pyramiidineuroneja hiiren neokorteksissa. Tässä työssä kyseistä neuronidataa analysoitiin statististen menetelmien avulla ja käytettiin laskennallisen neuronikasvumallin optimointiin. Hiiren aivoista saadun datan ja laskennallisella mallilla saatavien neuronirakenteiden eroja tutkittiin sekä vertailtiin aivojen eri osissa olevia eroja.

Kaksi L5 pyramiidineuronipopulaatiota merkittiin geneettisesti (etv, glt) vihreällä fluoresenssiproteiinilla. Populaatiot poimittiin somatosensoorisesta (barrel) korteksista ja visuaalikorteksista. Neuronien rakenteet rekonstruoitiin kolmiulotteisena mikroskoopin ja Neurolucida-ohjelmiston avulla (Groh et al 2010). Kolmiulotteinen data analysoitiin tässä työssä haarautumis- ja pituusominaisuuksien saamiseksi mallin optimisointia varten.

Netmorph-simulaattorilla (Koene et al 2009) voidaan tehdä malleja suurien pyramiidineuronien luomiseksi. Mallien parametreja voidaan optimoida stokastisilla menetelmillä. Menetelmäksi valittiin Retrospecive Approximation-optimointi jossa käytettiin lisäksi Nelder-Mead suorahakua. Optimoinnissa malli pyritään saamaan vastaamaan hiiren aivoista saatua dataa. Optimoinnin onnistumisen arviointiin käytettiin Kolmogorov-Smirnov testiä.

Optimoiduilla Netmorph-malleilla saatiin luotua realistia neuronirakenteita jotka vastasivat myös statistisesti tietyin osin hiirestä saatuja näytteitä. Suurien neuroniverkkojen luominen olisi mahdollista saatujen tulosten avulla ja niiden elektrofysiologisia ominaisuuksia voitaisiin tutkia esimerkiksi NEURON-ohjelmistolla simuloiden.

Avainsanat: Laskennallinen neurotiede, neuroanatomia, dendriittien morfologia, neokorteksi, pyramiidineuroni, Netmorph, stokastinen optimointi, Retrospective Approximation, Nelder-Mead, Kolmogorov-Smirnov

AALTO UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY

ABSTRACT OF THE
MASTER'S THESIS

Author: Risto Laakso

Title: Algorithmic description of pyramidal neuron populations from mouse neocortex

Date: 29.10.2010 Language: English Number of pages:6+61

Faculty of Electronics, Communications and Automation

Department of Biomedical Engineering and Computational Science

Professorship: Computational Engineering Code: S-114

Supervisor: Professor Jouko Lampinen

Instructor: Assistant Prof. Patrik Krieger

Large pyramidal neurons from the mouse neocortex have been studied by Groh et al (2010). In this thesis the neuronal data was statistically analyzed for structural features, and used to optimize computational models to create realistic neuronal morphologies. The mouse data and model generated morphologies were studied for populational differences in cortical areas.

Two populations of layer 5 pyramidal neurons were genetically labelled with enhanced green fluorescent protein and extracted from the somatosensory (barrel) cortex and the visual cortex in a previous study by Groh et al. The neuronal morphologies were reconstructed in 3D using microscope and Neurolucida software. The 3D data was analyzed in this thesis for the branching and length features needed for model optimization.

Computational models were constructed in the Netmorph simulator to generate large pyramidal neurons. The model parameters were optimized with the Retrospective Approximation method with a Nelder-Mead direct search to generate statistical distributions similar to the mouse data. The model optimization was evaluated with Kolmogorov-Smirnov goodness-of-fit tests.

The optimized Netmorph models were able to generate realistic and statistically similar morphologies to the experimental data. Large-scale networks could also be constructed and used for subsequent electrophysiological simulation in e.g. NEURON software.

Keywords: Computational Neuroscience, Neuroanatomy, Dendrite morphology, Neocortex, Pyramidal cell, Netmorph, Stochastic optimization, Retrospective Approximation, Nelder-Mead, Kolmogorov-Smirnov

Preface

The work presented in this thesis has been done in the Nobel Institute for Neurophysiology, Department of Neuroscience, Karolinska Institutet, Stockholm.

I would like to thank my supervisor Prof. Jouko Lampinen for his support in writing the thesis, and my instructor Assistant Prof. Patrik Krieger for providing the opportunity and for his constant help and guidance with neuroscience and the thesis in general.

I would also like to thank Mr. William Martin for proofreading my thesis, my family and friends for their support, and my girlfriend Daniella for her patience and caring during the process.

Stockholm, 20th September 2010

Risto Laakso

Contents

1 Introduction

The structure of neurons or their morphology has been studied since the pioneering work of Santiago Ramón y Cajal in the 19th century. The variety in neuronal morphologies is large and their functionality in networks is still mostly unknown. Recent increase in the processing power of computers has allowed construction of computational models to generate neuronal morphologies and to virtually simulate their activity. These new tools can be used to extend the study of large-scale neuronal networks.

1.1 Research objectives

The research objective was to see if computational algorithms can be used to generate realistic neuronal morphologies and to generate a large-scale network of neurons in some cortical area. The generated neuron morphologies should be statistically similar to experimental data.

The research objectives were summarized as:

O1 Can we use computational algorithms to simulate growth of L5 pyramidal neurons?

O2 Are the generated neurons statistically similar to experimental data?

O3 Can we create a large scale network from the generated neurons and simulate neuronal activity?

1.2 Limits of research

The research is limited to the data available or to pyramidal L5 neurons from mouse neocortex. The data is from the visual cortex and somatosensory (barrel) cortex. The electrophysiological characteristics are considered briefly, but the main focus is on structural properties.

2 Previous Work

2.1 The nervous system

The nervous system of humans and animals functions to detect changes in external and internal environment and to respond appropriately. It can be divided into the central nervous system (CNS) and to the peripheral nervous system (PNS). The basic structural and functional unit of the nervous system is the nerve cell or neuron.

The central nervous system consists of the brain and spinal cord, and contains the majority of nerve cell bodies. The peripheral system links the CNS to the periphery of the body, from which it receives information and transmits controlling impulses. The CNS is enveloped inside the meninges which consists of three layers: the dura mater, the arachnoid mater and the pia mater. The primary function of the meninges is to protect the central neural system.

Neurons receive information from sensory receptors or other neurons, and transmit information to other neurons or effector organs.

Various types of neurons exist, with different shapes and sizes. All neurons have a single cell body or soma, from which various branching processes extend. Most of the branching processes are receptive and are called the dendrites. One is transmittive and is called the axon.

Some regions of the CNS consist mostly of nerve cell bodies and are referred to as grey matter. Other regions contain mostly nerve processes, e.g. axons, and are called white matter. Nerve processes sharing common functions and connections tend to follow same pathway. [12]

2.2 Anatomy of the brain

The brain has two cerebral hemispheres, that have a highly convoluted outer mantle of grey matter and an inner core of white matter. Certain surface convolutions subserve specific sensory, motor or association functions. The cerebral cortex is a part that controls important sensory and motor functions. It includes the motor cortex, the somatosensory cortex, the visual cortex and the auditory cortex.

The cerebral cortex consists of the neocortex and the allocortex. The neocortex contains complex networks of neurons that receive input from the thalamus and from other cortical areas. Neurons in the neocortex connect to other cortical areas as well as the thalamus and other subcortical structures. [12] [39]

2.2.1 Cortical layers

The different regions of the neocortex have many similarities. The sensory cortices have six layers and are organized in columns, where neurons respond to the same sensory input. The thickness of the different layers varies in different regions of the neocortex, but it is always possible to identify six layers. Extensive connections exist between neurons in the different layers of the cortex.

Specific cell types are located in different layers of the cortex. Layer I is composed mostly of glial cells and axons traveling laterally, layer II has small granular cells, layer III has small pyramidal cells, layer IV has small stellate cells, layer V has giant pyramidal cells and layer VI has spindle or fusiform cells. [39]

2.2.2 Visual cortex

The visual cortex processes perceived visual information. It is organized in a similar way as other sensory cortices with six layers and vertical columns. These columns are organized in hyper-columns that form the functional units of a region of the visual field. [39]

2.2.3 Barrel cortex

The barrel cortex is part of the somatosensory cortex in rodents. It receives and processes information derived from the rodents facial whiskers. The barrel cortex contains six layers, of which layer IV is the main input layer. The cortex is named after distinctive barrel patterns that can be seen in layer IV, which replicate the pattern of whiskers on the face of the animal. Each whisker of the animal corresponds to a single barrel in the cortex. The topological position of barrels in the cortex is the same as the corresponding whiskers. [18]

2.3 Neuron types

The neocortex contains two basic types of neurons, spiny dendrites (the stellate and pyramidal cells) and smooth dendrites (smooth cells). The pyramidal cells form about 70% of the neurons and smooth cells form about 20% of the neurons. The spiny cells are excitatory, whereas the smooth neurons are inhibitory.

The pyramidal cells constitute about two-thirds of the neurons in the neocortex. They are found in all cortical layers except layer I. They are characterized by a long dendrite called the apical dendrite, which extends from the cell body or soma towards the pia mater. The other dendrites extending from soma are called the basal dendrites. Some dendrites can also branch from the apical; these are called oblique dendrites, except near the pia mater where they are called the tuft.

Neurons are connected to each other by synapses, small contact sites that allow neurons to transmit and process information. A synapse converts a presynaptic electrical signal into a chemical signal in the transmitting neuron, and back into a postsynaptic electrical signal in the receiving neuron. [53]

2.3.1 Nerve cell density

The number of neurons per unit volume varies a great deal. The density is lower in larger brains, which have on average larger sized neurons. The neural processes are longer in larger brains, but thickness stays mostly the same. The density of neurons in mouse cortex has been estimated as $9.2 * 10^4/mm^3$. The total number

of neurons in the mouse cortex has been estimated as $1.6 * 10^7$ in both hemispheres. The number of neurons in the human cortex has been estimated as $1 * 10^{10}$.

The density of synapses has been estimated as $7.2 * 10^8 / mm^3$ in the mouse cortex. The ratio of synapses to neurons has been estimated as $7826 : 1$, or approximately 8000 synapses per neuron. Axons have approximately one synapse for every $5\mu m$, and dendrites two synapses for $1\mu m$. The majority of excitatory synapses in the cortex connect one pyramidal cell to another pyramidal cell (75%). The range of axons in pyramidal cells is about $1mm$, with an axonal tree length of $10 - 40mm$. The range of dendrites can be about $0.2mm$ or longer, with a tree length of $4mm$. [7]

2.3.2 Electrophysiology

Neurons receive and transmit information by chemical and electrical means. The internal function of a neuron can be described electrically. Rall presented [45] [47] a one-dimensional cable theory for dendrites, where the dendritic tree is decomposed into a set of inter-connected cylindrical cables. The electric current can flow axonally inside the dendrite, or it can flow into the membrane. If the dendrites are assumed passive, i.e. that all parameters are time- and voltage-independent, the flow of current can be described by a one-dimensional passive-cable equation:

$$\left(\frac{r_m}{r_i}\right) \frac{\partial^2 V(x,t)}{\partial x^2} - r_m c_m \frac{\partial V(x,t)}{\partial t} - V(x,t) = 0, \tag{1}$$

where $r_i[\frac{\Omega}{cm}]$ is axial resistance, $r_m[\Omega cm]$ membrane resistance, and $c_m[\frac{F}{cm}]$ membrane capacitance. The general solution can be expressed as an infinite sum of decaying exponentials:

$$V(x,t) = \sum_{i=0}^{\infty} C_i e^{-t/\tau_i}, \tag{2}$$

where C_i depends on point of observation (x) and the input current, but is independent of time (t).

The cable-model was later extended by Rall [46]. The continous equation is discretized into a finite set of compartments, where each is a lumped representation of a small section of the dendritic tree. The current flowing in compartment j is then:

$$\frac{d}{4\hat{r}_j} \frac{V_{j+1} - 2V_j + V_{j-1}}{\Delta x^2} - \hat{c}_{m_j} \frac{dV_j}{dt} - i_{ion_j} = 0, \tag{3}$$

where d is compartment diameter, $\hat{r}_j$ is axial resistance, $\hat{c}_{m_j}$ is membrane capacitance of the j:th compartment, and i_{ion_j} is total ionic current that flows through the compartment membrane. [60]

2.4 Algorithmic description of neuronal structure

Various algorithmic descriptions of neuronal structures have been studied. Neuronal tree growth is usually considered as events of branching and elongation. In branching or bifurcation, some segment of the tree will branch into two new segments. In elongation, a segment will grow in length by some amount. At the end of the algorithm the ending segments are called terminal segments. Most of the algorithms are stochastic processes where the branching and termination probabilities are determined by experimental data. In the following sections algorithms are briefly reviewed.

2.4.1 Hillman, Tamori

Hillman suggested [24] that a small set of fundamental parameters could be sufficient to describe the structure of a neuron. In Hillman's algorithm, a dendrite would start with an initial diameter, and would thereafter elongate for some length and taper (i.e. diminish in diameter). After elongation, the dendrite would branch if the diameter was above some threshold value, or otherwise elongate more until termination. In case of branching, the daughter diameters would be determined by Rall's power rule [45].

Tamori extended [61] Hillman's approach by considering the effective volume of dendritic segments, which is then used to compute angles for new branches.

2.4.2 Burke

Burke suggested [10] a stochastic Monte Carlo method to simulate dendritic growth. The stochastic growth for each segment would start with some selected start diameter d_s and elongation rate ΔL. The segment could then branch or terminate. The branching and termination was controlled by probabilities P_b and P_t, which were estimated from observed experimental data. At each step, Monte Carlo simulation was done to determine if branching or termination should occur. If neither were to occur, the segment would be elongated by ΔL and tapered by some taper rate. The process would then be repeated until termination.

2.4.3 L-neuron

Ascoli et al [2] presented L-neuron software, which uses algorithms similar to those of Hillman, Tamori and Burke to construct realistic neuronal morphologies. The L-neuron software is based on Lindenmaeyer L-systems which are procedural algorithms that can generate realistic branching structures of trees and plants. A set of values for the "fundamental parameters" can be specified, and the L-neuron system will generate realistic neuronal morphologies from these. An L-measure program was also introduced to measure the required parameters from experimental data.

2.4.4 ArborVitae

Senft et al [51] presented the ArborVitae software, which takes a more holistic approach. Instead of modeling single dendrites, ArborVitae considers populations of neurons. The algorithm constructs groups of neurons at the same time, trying to achieve realistic morphologies and layout in 3D-space.

2.4.5 Samsonovich

Samsonovich et al [49] present a stochastic algorithm to generate dendritic morphologies. The algorithm works in two phases, first generating the dendritic tree structure and then assigning diameters for each segment. The bifurcation probability at each step is sampled from a probability distribution with distance to soma as a parameter. The distributions were determined by fitting them to experimental data. Different distributions were used for different neuronal classes and for basal/apical dendrites. The segment diameters were derived similarly by sampling probability distributions.

2.4.6 Van Pelt

Van Pelt et al have studied probabilistic branching models for dendrites, see for example [66] [67] [69]. Various branching models have been propsed. The models are summarized below:

QS-model The branching probability of intermediate and terminal segments is considered in the QS-model, where S regulates branching probability as a distance from soma, and Q regulates branching probability of intermediate segments. The branching probability p_{term} of the terminal segment at a centrifugal distance γ is determined by $p_{term} = C_1 2^{-S\gamma}$, where C_1 is the normalization constant and parameter S the regulation constant. For $S = 0$ all terminal segments have equal probability of branching, and for $S = 1$ the probability will decrease as a factor of two for each following order (distance from soma). The probability of intermediate segment branching is $p_{int} = \frac{Q}{1-Q} p_{term}$, with $Q \in [0, 1)$. For $Q = 0$ there are no intermediate segment branching, and for $Q = 1$ only intermediate segments branch. The authors then argue that based on experimental data, it seems that only terminal segment branching is occuring, and therefore Q should be 0.

B-model A discrete-time stohastic process was considered where the dendrites grow for some time T, which is divided into short timebins i with durations T_i. For each timebin a branching probability is defined. In B-model, a constant branching probability is considered as

$$p_i = \frac{B}{N}, \tag{4}$$

where B is the number of branches of a single path during the entire simulation, and $N = \frac{T}{\Delta T}$ is the total number of steps in the simulation. A Monte Carlo simulation can then be performed, where at each step segments have a probability p_i of branching.

BE-model The B-model is extended to allow degree-dependent probabilities. The branching probability function is

$$p_i = \frac{B}{Nn^E}, \tag{5}$$

where B is as above, n is the current number of terminal segments in the tree, and $E \in [0, 1]$ is constraining factor. For $E = 0$ the probability is reduced to that of B-model. For $0 < E \leq 1$ each additional branching will reduce the probability of branching. This can be seen as a competition for growth resources.

BES-model In BES-model an additional parameter S is introduced from the QS-model. The probability is

$$p_i = \frac{C2^{-S\gamma}B}{Nn^E}, \tag{6}$$

where S is the constraining factor depending on centrifugal order γ. The C is the normalization factor $C = n/\sum_j^n 2^{-S\gamma_j}$ to keep the total probability the same.

2.4.7 Netmorph

Koene et al [32] introduced Netmorph, a framework to generate networks of neurons with realistic morphologies. The framework uses a discrete stochastic simulation to generate neural morphologies for one or more neurons. It includes some alternative models for each function, mainly those of van Pelt. Most parameters are specified as probability distributions, which are then sampled during the simulation.

2.4.8 CX3D

Zubler et al [78] recently introduced a framework to simulate neuronal growth. The framework uses molecular simulation to model physical, chemical and biological interaction between objects. The framework can be freely downloaded from http://www.ini.uzh.ch/projects/cx3d/. The ability to simulate growth of some neurons has been demonstrated.

2.5 Branching processes

The process of dendritic growth can also be regarded as a branching process. Branching processes were originally studied as part of population dynamics, when Galton and Watson wanted to study the extinction of family names. The Galton-Watson

branching process is defined as follows: some initial population of Z_0 individuals is considered, and then subsequent generations $Z_1, Z_2, \ldots, Z_n$. At each generation, every individual i can produce X_i offspring, where X_i is a random variable. The individuals have equal probability of reproduction and they reproduce independently of each other. The number of offspring for the individual i is determined by some probability distribution $P(X_i = k) = p_k$, where p_k is the probability of having k children. The population size at generation Z_{n+1} is then

$$Z_{n+1} = \sum_{i=0}^{Z_n} X_i, \tag{7}$$

i.e. the number of offspring by the previous generation. The subsequent population sizes depend only on the previous generation, and therefore form a Markov chain. [23] [15]

The expected number of offspring by individual is

$$E[X] = \mu = \sum_{k=0}^{\infty} k p_k. \tag{8}$$

If the initial population $Z_0 = 1$, then the expected population size for generation n is

$$E[Z_n] = \mu^n, \tag{9}$$

and variance

$$Var[Z_n] = \frac{\sigma^2 \mu^{n-1}(\mu^n - 1)}{\mu - 1}, \qquad \mu \neq 1, \tag{10}$$

where σ^2 is the variance of Z_1, $\sigma^2 = E[Z_1^2] - \mu^2$. [23] [15]

In case that the number of offspring depends on total population size, the analysis is harder. It has been studied by e.g. Jagers[28], and Jagers and Klebaner[29].

2.5.1 Dendrite branching processes

Branching processes have been studied in biological problems [31], and Uemura et al [64] have studied the application of random binary splitting branching process in neuroscience. The work of Uemura et al is briefly mentioned by van Pelt et al in [70].

The dendritic growth process is considered here as a branching process, slightly different from that of Uemura et al [64]. Considering van Pelt's BE-model, let generation n be the n:th step in the discrete simulation process. The dendrites can grow from terminal segments, so the population size Z_n is then the number of terminal segments at step n. At each step segment can branch, giving two new terminal segments, or not branch, where it continues intact to the next generation. Let p_b be the branching probability. The offspring probabilities can then be considered as $P(X = 1) = (1 - p_b)$ for not branching, $P(X = 2) = p_b$ for branching, and

$P(X = k) = 0, k \notin \{1, 2\}$ for all other cases. The expected number of offspring is then

$$E[X] = \mu = \sum_{k=0}^{n} kp_k = 1(1 - p_b) + 2(p_b) = 1 + p_b. \tag{11}$$

In the BE-model the branching probability $p_b = \frac{B}{Nn^E}$ is dependent on the population size n when $E > 0$. For $E = 0$ the expected population size can be computed for the n:th generation:

$$E[Z_n] = \mu^n = (1 + p_b)^n = (1 + \frac{B}{N})^n. \tag{12}$$

For the last generation, $n = N$, this becomes:

$$E[Z_N] = \mu^N = (1 + p_b)^N = (1 + \frac{B}{N})^N. \tag{13}$$

If the mean number of terminal segments μ_{TS} is known, parameter B can be computed:

$$\begin{aligned} \mu_{TS} &= (1 + \frac{B}{N})^N & (14)\\ \frac{B}{N} &= \mu_{TS}^{1/N} - 1 & (15)\\ B &= N(\mu_{TS}^{1/N} - 1). & (16) \end{aligned}$$

If the difference between two generations, $\Delta Z = Z_{n+1} - Z_n$ is considered,

$$\begin{aligned} E[\Delta Z|Z_n] &= E[Z_{n+1}|Z_n] - Z_n & (17)\\ &= \mu Z_n - Z_n = (1 + p_B)Z_n - Z_n & (18)\\ &= Z_n p_B & (19) \end{aligned}$$

Then, for $p_B = \frac{B}{NZ_n^E}$,

$$\Delta Z = Z_n p_B = \frac{Z_n B}{NZ_n^E} \tag{20}$$

If $E = 0$,

$$\begin{aligned} \Delta Z &= \frac{Z_n B}{NZ_n^0} = \frac{Z_n B}{N} & (21)\\ Z_{n+1} &= Z_n + \Delta Z = Z_n + \frac{Z_n B}{N} = Z_n(1 + \frac{B}{N}) & (22)\\ Z_{n+2} &= Z_{n+1}(1 + \frac{B}{N}) = Z_n(1 + \frac{B}{N})(1 + \frac{B}{N}), & (23) \end{aligned}$$

giving $Z_k = (1 + \frac{B}{N})^k$, as above ($Z_0 = 1$). If $E = 1$,

$$\Delta Z = \frac{Z_n B}{N Z_n^1} = \frac{B}{N} \tag{24}$$

$$Z_{n+1} = Z_n + \Delta Z = Z_n + \frac{B}{N} \tag{25}$$

$$Z_{n+2} = Z_{n+1} + \frac{B}{N} = Z_n + \frac{B}{N} + \frac{B}{N}, \tag{26}$$

giving $Z_k = 1 + k * \frac{B}{N}$, or $Z_N = 1 + B$. For other values of E, we get a recursive formula:

$$\Delta Z = \frac{Z_n B}{N Z_n^E} \tag{27}$$

$$Z_{n+1} = Z_n + \Delta Z = Z_n + \frac{Z_n B}{N Z_n^E} \tag{28}$$

$$Z_{n+2} = Z_{n+1} + \frac{Z_{n+1} B}{N Z_{n+1}^E} = Z_n + \frac{Z_n B}{N Z_n^E} + \frac{(Z_n + \frac{Z_n B}{N Z_n^E})B}{N(Z_n + \frac{Z_n B}{N Z_n^E})^E} \tag{29}$$

2.6 Neuronal statistics

Various statistical properties can be computed for the neuronal trees. The statistics presented here are grouped by their categories. The neuronal tree is first defined as a graph structure.

2.6.1 Neuronal tree as a graph

A graph is a pair $G = (V, E)$ of a finite set V and set E of pairs (a, b) where $a \neq b$ are elements of V. In the directed graph the pairs in set E are ordered. The elements of V are called vertices and the elements of E are called edges. An edge e_{ij} connects two vertices v_i and v_j. [30]

A neuronal tree can be considered as a graph. The tree root, all branching points and all endings are the vertices V. If two points are connected in the neuronal tree, then an edge is formed between them. In addition, the edges are directed so that they point away from the root vertex. For vertex v_i, all edges e_{ai} are arriving edges (pointing to vertex v_i), and all edges e_{ib} are leaving edges (pointing from vertex v_i).

We can then define root, branching, continuation and terminal points. The root vertex is the vertex with no arriving edges. The branching points are the vertices with one arriving edge and two leaving edges. The continuation points are the vertices with one arriving edge and one leaving edge. The terminal points are the vertices with one arriving edge and no leaving edges.

2.6.2 Topological properties

The topological properties of neurons can be described with the number of segments or their relative positions.

Number of endings/terminal segments is defined as the number of terminal points/vertices.

Number of branching points/intermediate segments is defined as the number of branching points/vertices.

Centrifugal order is defined as the mean topological distance from root to segments, i.e. the topological distance from root to current segment is computed for all segments and averaged [70].

Asymmetry index is defined as the symmetry of the tree [68] [70]. For all branching points, the number of endings is counted in the left subtree (l) and the right subtree (r). The *partition asymmetry* is then

$$A_i^{part} = \frac{|l - r|}{l + r - 2} \tag{30}$$

for some branching point v_i, $A^{part}(1, 1) = 0$. The *asymmetry index* is then

$$A^{tree} = \frac{1}{n} \sum_{i=0}^{n} A_i^{part}. \tag{31}$$

2.6.3 Metrical properties

The length of segments can be described with metrical properties.

Total length is defined as the sum of lengths for all segments in the tree.

Intermediate length is defined as the sum of lengths for all intermediate segments.

Terminal length is defined as the sum of lengths for all terminal segments.

2.6.4 Spatial properties

The position of segments in 3D-space can be described by spatial properties.

Sholl analysis is a spatial analysis of branching point locations [54]. The *Sholl sequence* is a sequence indicating the number of branching points within distances r_i from the cell body. Initially a circle C_{r0} with radius $r = r_0$ is drawn around the cell body, and the number of branching points within the radius r_0 are counted. Then the radius is increased, $r_1 = r_0 + \Delta r$, and the number of branching points contained in C_{r1} but not in C_{r0} are counted. This is continued for circles of increasing radiuses until all branching points have been encountered.

2.7 Statistical comparison and hypothesis testing

2.7.1 Analysis of Variance (ANOVA)

Multiple groups of samples can be tested for statistical equality with ANOVA. ANOVA will test if the groups have equal means. One-way ANOVA will test for differences in two or more independent groups. The distributions are also assumed to be normal and have equal variances or homoscedasticity. The null hypothesis is that all the groups are drawn from the same population.

To perform one-way ANOVA the equality of variances in groups can be tested with Levene's test. Levene suggested doing one-way ANOVA of the absolute deviations of the means [35]:

$$z_{ij} = |y_{ij} - \bar{y}_i|,\ i = 1, 2, \ldots, t,\ j = 1, 2, \ldots, n_i. \tag{32}$$

Brown and Forsythe extended Levene's test for non-normal cases by using sample medians instead of means, $z_{ij} = |y_{ij} - y_{imed}|$.[8] [38] [55]

In repeated measures ANOVA the sample groups can be tested for equal variance by a sphericity test. Mauchly introduced the sphericity test in [37]. ANOVA can be used for non-equal variances by using an correction factor to the degress of freedom. Possible choices include Greenhouse-Geisser [20] and Huynh-Feldt [27] corrections.

2.7.2 Student's t-test

Student's t-test can be used to test for the null hypothesis that means of two normally distributed populations are equal. If the test is performed multiple times a correction can be introduced. Choices include Bonferroni-, Holm-Bonferroni-, Šidák- and Dunnett's corrections. [6] [38] [34]

Bonferroni correction adjusts alpha by number of tests, using

$$p \leq \frac{\alpha}{s}, \tag{33}$$

where s is the number of tests.

2.7.3 Kolmogorov-Smirnov test (K-S test)

Kolmogorov-Smirnov test can be used to test for equality between non-normal distributions. The K-S test compares the distance between two distributions. The null hypothesis is that the samples are drawn from the same distribution. The test can be used as a goodness-of-fit test between observed and generated distributions. [33] [56] [36] [62]

2.8 Model optimization

Neuronal growth models generate neuron structures. The statistical properties of the generated neurons depend on some model parameter vector $\mathbf{x}$. To get neurons statistically similar to experimental data, we want to find the optimal parameters $\mathbf{x}^*$, so that the model output $f(\mathbf{x}^*) = \mathbf{y}^*$ corresponds to the experimental data. If $\mathbf{x}^*$ is not known, some parameter vector $\mathbf{x^k}$ can be tried and the model error e

$$e(\mathbf{x^k}) = f(\mathbf{x}^*) - f(\mathbf{x^k}) = \mathbf{y}^* - \mathbf{y^k} \tag{34}$$

is computed. The optimal parameter vector is then a vector which minimizes $e(x)$.

Optimal parameters can be found in various ways. One of the simplest is to start with some vector $\mathbf{x^0}$ and searching for better choices in the neighbourhood of $\mathbf{x^0}$. The best choice is selected and search is performed again around the new choice, until an optimal choice is found or some pre-determined iteration count is reached. These methods are called direct seach methods and are discussed in Section 2.8.1.

If the function $f(x)$ can be differentiated the derivate can be used to find optimal parameters, usually much faster than with direct search. One such method is called Newton's method and it is discussed in Section 2.8.2.

For functions where the function value is generated by random process stochastic methods can be used. Stochastic optimization is discussed in Section 2.8.3.

2.8.1 Direct search

Direct search methods look for optimal parameter vectors by searching for a better parameter vector which differs only slightly from the current best choice.

The Nelder-Mead method is a direct search method that proceeds by forming a simplex in the parameter space. For N dimensional space N+1 points are chosen to form a simplex (i.e. for two dimensional space, we have a simplex with three points or a triangle). The function is evaluated at each end-point of the simplex and the points are sorted by function value. At each iteration, the worst point of simplex is considered for removal. First a reflection point is computed by reflecting the worst point against a line formed by best and second-best points. If this does not provide any improvement, a contraction point is tried, where the worst point is moved closer to the best and second-best points. If neither improves the worst point the simplex is shrunk by moving all but the best point closer to the best point. [40] [41]

2.8.2 Newton's method

When the function $f(x)$ can be differentiated, the derivate $f'(x)$ can be used to determine which way x should move to achieve possibly optimal parameters. In addition, if the function is twice differentiable, the second derivate $f''(x)$ can be used. [41] [44] [17]

For vector parameters the gradient $\nabla f(x)$ can be computed by differentiating

$f(x)$ against each parameter, i.e.

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \frac{\partial f}{\partial x_2}(\mathbf{x}), \ldots, \frac{\partial f}{\partial x_n}(\mathbf{x}) \right). \tag{35}$$

If $f(x)$ is multi-valued, $f(x) : R^m \rightarrow R^n$, the gradient can be computed for each output value, giving the Jacobian matrix

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \nabla F_1 \\ \nabla F_2 \\ \vdots \\ \nabla F_n \end{bmatrix} = \begin{bmatrix} \frac{\delta F_1}{\delta x_1} & \frac{\delta F_1}{\delta x_2} & \cdots & \frac{\delta F_1}{\delta x_m} \\ \frac{\delta F_2}{\delta x_1} & \frac{\delta F_2}{\delta x_2} & \cdots & \frac{\delta F_2}{\delta x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta F_n}{\delta x_1} & \frac{\delta F_n}{\delta x_2} & \cdots & \frac{\delta F_n}{\delta x_m} \end{bmatrix} \tag{36}$$

If $f(x)$ is twice differentiable, the second derivate or the Hessian can be computed

$$\mathbf{H}(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial^2 x_n^2} \end{bmatrix} \tag{37}$$

Steepest-Descent When the function is once differentiable, the gradient information can be used. In the steepest-descent method the parameter space is searched by moving the current vector $\mathbf{x}$ in the direction opposite of the gradient (or Jacobian),

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mu \nabla f(\mathbf{x}_k) \tag{38}$$

where μ is step size (usually $\mu \in [0, 1]$). This is continued until the gradient is zero or maximum iteration count is reached. The step size μ can be fixed constant or all values within some range can be searched for an optimal value of μ. [41] [44] [17]

Quadratic model Many optimization methods work well when used for quadratic functions. [41] [44] [17] General functions are also often well approximated with quadratic expansion around a local minimum point. For twice differentiable functions a Taylor series expansion of $f(x)$ gives

$$f(x + \delta) \approx f(x) + \delta^T \nabla f(x) + \frac{1}{2} \delta^T \nabla^2 f(x) \delta + \ldots \tag{39}$$

the minimum can then be found by iteration

$$\delta = -\nabla^2 f(x)^{-1} \nabla f(x) \leftrightarrow \delta = -H^{-1}(x) \nabla f(x) \tag{40}$$

$$x_{k+1} = x_k + \delta \tag{41}$$

Quasi-Newton method Computation of the second derivate, $\nabla^2 f(x)$ or the Hessian, can be difficult or time consuming. In Quasi-Newton methods [41] [44]

[17] the Hessian is approximated at each step,

$$\begin{aligned} \delta &= -\hat{\mathrm{H}}_k^{-1}(\mathbf{x}_k)\nabla f(\mathbf{x}_k) && (42)\\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \mu\delta && (43)\\ \hat{\mathrm{H}}_{k+1} &= g(\hat{\mathrm{H}}_k) && (44) \end{aligned}$$

One choice for updating the Hessian is BFGS-formula by Broyden [9], Fletcher [16], Goldfarb [21] and Shanno [52],

$$\mathrm{H}_{k+1}^{\mathrm{BFGS}} = \mathrm{H}_k + \left(1 + \frac{\gamma^T H_k \gamma}{\delta^T \gamma}\right)\frac{\delta\delta^T}{\delta^T\gamma} - \left(\frac{\delta\gamma^T H_k + H_k\gamma\delta^T}{\delta^T\gamma}\right). \quad (45)$$

Newton-Raphson method Another possibility is to use the Newton-Raphson method [41] [44] [17] which only utilizes the first derivate. If the function $f(x)$ reaches zero at the minimum point then the minimum can be found by repeat evaluations

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{f(\mathbf{x}_k)}{\nabla f(\mathbf{x}_k)}. \quad (46)$$

2.8.3 Stochastic optimization

When the function value is nondeterministic or different for subsequent evaluations with same input values, e.g. as a result of a random process, most optimization methods will perform badly. One possibility is to evaluate the function multiple times and compute the sample average at each point. The optimization can then be performed by using the sample average value in place of the function value. This can work if the function variance is small or the function can be sampled enough times.

Direct search methods can be used almost as in the deterministic case, by utilizing sample averages. Tomick et al [63] have proposed modifications to Nelder-Mead for stochastic optimization. In a study by Barton et al [5] two modifications, S9 and RS, were statistically significant in helping Nelder-Mead perform better. In S9 the shrink amount, δ is reduced from the original value of 0.5 to 0.9. In RS the best point is sampled again in shrinking, for the possibility that the best value is only best because of a sampling artifact.

Robbins and Monro argued [48] that averaging a function over multiple samples is wasteful if the result is only used for intermediate step in optimization. The optimization should proceed by using initially a low number of samples and then increasing the number of samples as the algorithm gets nearer to the solution. Recent methods extending the idea of Robbins and Monro include SPSA, Retrospective approximation and SMRAS.

SPSA Spall has proposed simultaneous perturbation stochastic approximation (SPSA) method [57] and Adaptive SPSA method [58], [59]. The algorithms minimize the loss function $L(\theta)$ with the gradient descent method by using two-sided

finite-difference (FD) approximation to the gradient function, where the function is evaluated at random locations $\theta_k + c_k\Delta_k$ and $\theta_k - c_k\Delta_k$. Instead of evaluating the FD approximation components separately for each dimension, the SPSA approach evaluates it in "all" dimensions at once. The locations are selected by the random direction vector Δ_k, which has to satisfy some constraints. One choice is to use symmetric Bernoulli (± 1) distribution to generate the vector components. The optimization then proceeds as in Newton's method, with diminishing step size by each iteration step.

Retrospective Approximation Chen et al presented [11] a retrospective approximation (RA) method to stochastic root finding. The method proceeds by finding successive approximative roots for the stochastic function by using an increasing number of samples to compute function value averages. At each step, some direct search method is used to find an approximative root for the function. The search will use sample averages of N points to evaluate the function at each point. After finding some approximate root, the point is used as an initial point for the next search, and the number of samples used for averages is increased. Initially the search will only use fast, low-quality estimates to find an approximate root, and when the search progresses, the estimates will become slower and more accurate. The final estimate is produced after K iterations by computing a weighted average of the approximate roots, where each root is weighted by the number of samples used during its computation.

SMRAS Hu et al have presented the model reference adaptive search (MRAS) [25] and Stochastic MRAS (SMRAS) methods [26]. The methods use a family of parameterized probability distributions on the solution space, which are sampled to generate candidate solutions. The candidate solutions are evaluated and the information is used to update the distribution parameters. The updated distributions are then sampled again, until a solution is found.

3 Research data and methods

Neuronal data was extracted from transgenic mice and reconstructed in 3D (3.1). A computational model was configured and optimized to statistically match the model with the mouse data (3.2-3.4).

3.1 Research data

Neuronal data was extracted from a transgenic mouse population. The mice were genetically modified to mark ETV or GLT-expressing neurons with fluorescent green color protein (EFGP). The ETV-expressing neurons reside in the layer 5a in the cerebral cortex and the GLT-expressing neurons in layer 5b. The fluorescent marking allows identification of the same neuronal population in various mice, allowing accurate categorization. [22]

The neurons were extracted by performing a histology, where the brain is sliced to thin slices (approx $100\mu m$ thick). The slices go through chemical processing which shrinks the tissue slighly. After processing the slices are observed with a microscope. Neurons visible in the slices are traced using microscope and Neurolucida software [19]. The microscope is focused at some depth in the slice and the neurons are traced in a 2D-plane using Neurolucida. The depth focus is altered and the process repeated until all depths are traced from the slice. The final tracing is then computed by combining the traced 2D planes to create a 3D tracing.

The data was grouped by cortical area (BC, VC) and genetical population (ETV, GLT). Statistics were computed with custom-written Java software. Some results were arbitarily selected and verified to be equal to values given by Neurolucida Explorer. The data is presented in Table 1 below.

3.2 Methodological choices

3.2.1 Neuronal growth factors

The neuron samples were extracted from a mice population where various internal and external factors can affect neuronal growth. The purpose of this study is to focus on growth factors related to ETV and GLT populations, their genetic differences, and possible near-environment factors. Various hypotheses were formed to test for variation caused by other factors. Measurable factors that can be tested from the data are related to individual age and brain size, histology caused tissue shrinkage, and dendrite cutting in slicing.

3.2.2 Growth modeling framework

Various statistical methods to reconstruct neuronal morphologies were reviewed in Section 2.4. Most of these consider reconstruction of a single neuron based on measured statistical properties. One research objective in this work is to study the

Table 1: Summary of measurements for mouse data

	NUMBER OF SAMPLES			NUMBER OF DENDRITES (per neuron)		
	basal	apical	oblique	basal	apical	oblique
BC-ETV	205	29	111	7.07 ± 2.23	1	3.83 ± 2.36
BC-GLT	154	20	171	7.70 ± 2.43	1	8.55 ± 3.33
VC-ETV	156	20	85	7.80 ± 1.82	1	4.25 ± 1.59
VC-GLT	111	16	142	6.94 ± 1.53	1	8.88 ± 4.33

	NUMBER OF ENDINGS (per dendrite)			ASYMMETRIC INDEX (per dendrite)		
	basal	apical	oblique	basal	apical	oblique
BC-ETV	4.64 ± 3.43	19.86 ± 8.05	2.38 ± 1.89	0.13 ± 0.11	0.24 ± 0.06	0.07 ± 0.11
BC-GLT	3.44 ± 2.25	36.25 ± 10.59	1.92 ± 1.21	0.11 ± 0.12	0.26 ± 0.03	0.05 ± 0.09
VC-ETV	3.85 ± 2.49	19.35 ± 7.44	1.80 ± 1.09	0.13 ± 0.12	0.25 ± 0.04	0.05 ± 0.09
VC-GLT	3.89 ± 2.34	30.31 ± 9.73	1.53 ± 0.76	0.14 ± 0.11	0.25 ± 0.03	0.02 ± 0.06

	TOTAL LENGTH (μm, per dendrite)			INTERMEDIATE SEGMENT LENGTH (μm)		
	basal	apical	oblique	basal	apical	oblique
BC-ETV	421 ± 331	2343 ± 796	225 ± 202	20 ± 21	47 ± 103	27 ± 29
BC-GLT	335 ± 250	4802 ± 1390	182 ± 122	23 ± 22	52 ± 67	28 ± 25
VC-ETV	349 ± 265	2248 ± 661	187 ± 130	21 ± 24	47 ± 69	32 ± 31
VC-GLT	370 ± 289	3903 ± 1213	152 ± 99	20 ± 19	45 ± 61	29 ± 28

	TERMINAL SEGMENT LENGTH (μm)		
	basal	apical	oblique
BC-ETV	73 ± 44	68 ± 40	76 ± 46
BC-GLT	81 ± 46	82 ± 54	82 ± 47
VC-ETV	75 ± 48	72 ± 45	90 ± 51
VC-GLT	81 ± 51	86 ± 49	90 ± 49

reconstruction of large-scale neuronal networks. Netmorph [32] was selected as the modeling framework because of its capability to construct large-scale networks.

3.3 Research implementation

The experimental neuronal data was in Neurolucida .asc-files. The files can be viewed with Neurolucida Explorer and some statistics can be computed. The statistical analysis performed in this work, however, requires more statistics than provided by Neurolucida Explorer. A custom written software was therefore developed in Java to read the data files and to compute all the required statistical properties. The values were compared to those given by Neurolucida Explorer to verify correctness, in applicable parts.

The Netmorph framework has been developed in C++ and released by Koene et al [32]. The source code can be freely downloaded from http://netmorph.org. Version 20090224.1225 was compiled on both a 64bit Linux platform (Ubuntu 10.04, GCC 4.4.3) and Windows XP (SP2, Cygwin GCC 3.3.4).

Netmorph provides some statistics about the generated neurons but more were required. The custom written software was extended to read Netmorph-generated files and to compute the same statistics as for Neurolucida files. The software was also extended to export neuronal statistics to Excel spreadsheets, to export neuronal morphologies and statistics to Matlab files, and to export images of neurons to EPS and PNG formats.

The parameter optimization process was implemented in Java. Various optimization algorithms were implemented in Java for evaluation. Function evaluation

would create a Netmorph configuration file, run Netmorph and parse results. The Java matrix library (JAMA) was used for linear algebra.

The final choice was to use Retrospective approximation (RA) with a Nelder-Mead direct search for optimization. To make the optimization run faster Nelder-Mead was parallelized to run multiple function evaluations at the same time in each step (initialization step, reflection/expansion/contraction step, shrink step). Netmorph simulations were also run in parallel in a computing cluster, using the Java parallel processing framework (JPPF) for coordination and communication.

Statistical analysis was performed in R (www.r-project.org) and in Matlab (Mathworks). Data visualization was done by custom written software and in Matlab.

3.4 Used methods

3.4.1 Environmental factor determination

Measurable environmental factors that can cause variation in the data were identified as individual age, individual brainsize, tissue shrinkage in preparation, and dendrite cutting in preparation. To test for the presence of these factors hypotheses were formed. The hypotheses are presented below, grouped by factor. Correlation was tested with the Matlab function corrcoef, which computes the correlation coefficient and a p-value, assuming that the data is normally distributed.

Age factors When the brain develops the dendrites can branch and grow in length. To test for age variation the following hypotheses were formed:

H1. Number of endings grows with age.

H2. Intermediate segment length does not grow with age.

H3. Terminal segment length grows with age.

Brain size factors Individual brain size affects the layer thicknesses and perhaps the density of neurons. Neuroanatomical research [12] suggests that apical dendrites should always grow to the pia mater in L5 pyramidal dendrites. Individuals with larger brain size should have neurons further away from the pia mater and therefore longer apical dendrites. This extra growth needed for the apical dendrite might affect the growth of other dendritic parts. Neural density has been hypothesized to be affected by brain size.

To test for brain size caused variation, the following hypotheses were formed:

H4. Number of endings correlates with brain size.

H5. Intermediate segment lengths correlate with brain size.

H6. Terminal segment lengths correlate with brain size.

Tissue shrinkage The brain slices shrink visibly during processing. The compression is emprically observed to be higher in the (slice-) Z-direction than in the X,Y-directions. This affects at least spatial analyses, and maybe segment

lengths. The effect of compression was visualized by plotting the data in the X,Z and X,Y -planes.

Dendrite cutting Some dendrites can be cut when the brain is sliced. The cutting can only occur at the slice border, so it was hypothesized that dendrites which end near the border should be affected. The following hypothese were then formed:

H7. Dendrites ending near the border have on average shorter length.

H8. Dendrites ending near the border have on average larger diameter.

3.4.2 Netmorph model construction

The Netmorph model was constructed by creating a configuration file that includes all model parameters. These parameters are discussed below.

Environmental The growth space was divided into regions I-VI, corresponding to the cerebral layers I-VI. The layer widths were based on empirical data (see Table 2). Additionally, the pia border was specified by defining a sphere at location (x=0,y=0,z=-50000) with radius r=50000, to create a slightly curving border at z=0. All neurons were specified to be in layer Va (ETV) or Vb (GLT).

Table 2: Layer thicknesses used in Netmorph model.

Layer	I	II	III	IV	Va	Vb	VIa	VIb
Thickness (μm)	93	118	118	233	106	160	200	84

Growth model The Van Pelts model was used for branching, and the ERI and TSEM models for elongation. Growth competition for elongation, $\nu(t) = \nu_0 n(t)^{-F}$, was disabled by setting $F = 0$.

Basal dendrites The number of basal dendrites was specified to be 7 during optimization for all groups. This allowed the optimization algorithm to compute the exact number of samples used in statistics, while still being low enough to be a realistic amount.

Apical dendrite The apical dendrites trunk length was specified to be 570 ± 10 to match layer 5 (in ETV). The number of obliques was set to 7 during optimization, for the same reasons as with the basal dendrites.

3.4.3 Estimating initial parameter values

Netmorph contains default values for all parameters which can be used as initial values. To help the parameter search converge faster these values can be improved by estimating better values based on neuronal properties. The initial values are discussed below and summarized in Figure 3.

B,E,S-parameters Van Pelt et al [70] [72] provide a diagram from which B,E and S can be initially estimated by empirical observation. Additionally they provide a recursive equation for estimating the terminal segment distribution for given B,E -values:

$$P(n,i) = \sum_{j=0}^{n/2} P(n-j, i-1) \binom{n-j}{j} [p(n-j)]^j [1-p(n-j)]^{n-2j} \qquad (47)$$

where $P(n,i)$ is the probability of tree degree n at time i, $P(1,1) = 1$, and $p(n) = B/n^E$ is the branching probability.

Initial B,E values were solved from (47) by optimization methods. Quasi-Newton and Newton-Raphson methods (Sec. 2.8.2) were tried. The Newton-Raphson method performed better with test data and was subsequently used.

ERI-parameters The elongation related parameters can be approximated as follows: A single path in the three branches B times on average. The intermediate segments in the path can grow for approximately $t_g = \frac{T}{1+B}$ steps. If we know the average intermediate segment length l_{int} we can then compute

$$\hat{\mu}_{eri} = l_{int} * \frac{1}{t_g} = \frac{l_{int}(1+B)}{T}. \qquad (48)$$

Initial branch length The initial branch length can be left to default values or estimated as small, e.g. $\mu_{ini} = 0.1, \sigma_{ini} = 10$.

Number of basal dendrites, obliques The numbers of basal dendrites and oblique dendrites were determined from data statistics.

3.4.4 Parameter optimization

Netmorph model parameters need to be optimized to achieve the desired distributions of neuronal features. No optimization functionality is provided by Netmorph. Newton-Raphson with sample averages, Quasi-Newton with sample averages, SPSA and Retrospective Approximation (RA) were implemented and tested with Netmorph simulations. RA converged fastest in test cases and was subsequently used for all optimization. To make the optimization run faster the Nelder-Mead direct

Table 3: Netmorph parameters and initial value estimates.

	Initial value estimate			
Parameter	BC-ETV	BC-GLT	VC-ETV	VC-GLT
all_dendrites.B_inf	1.8076	1.5862	1.7854	1.9507
all_dendrites.E	0.0516	0.3192	0.3325	0.4389
all_dendrites.eri.PDF.mean	0.00003889	0.00004259	0.00003889	0.00003704
all_dendrites.eri.PDF.std	0.00004259	0.00004259	0.00004444	0.00003519
all_dendrites.tsem.branch.PDF.mean	0.1	0.1	0.1	0.1
all_dendrites.tsem.branch.PDF.std	10	10	10	10
pyr1.tuft.B_inf	4.3055	6.00	6.00	6.00
pyr1.tuft.E	0.3460	0.25	0.35	0.30
pyr1.tuft.eri.PDF.mean	0.00005000	0.00005185	0.00005926	0.00005370
pyr1.tuft.eri.PDF.std	0.00005556	0.00004630	0.00005741	0.00005185
pyr1.tuft.tsem.branch.PDF.mean	0.1	0.1	0.1	0.1
pyr1.tuft.tsem.branch.PDF.std	10	10	10	10
pyr1.oblique.B_inf	1.0678	0.7199	0.6510	0.5105
pyr1.oblique.E	0.0000	0.2080	0.2432	0.7311
pyr1.oblique.eri.PDF.mean	0.00005000	0.00005185	0.00005926	0.00005370
pyr1.oblique.eri.PDF.std	0.00005556	0.00004630	0.00005741	0.00005185
pyr1.oblique.tsem.branch.PDF.mean	0.1	0.1	0.1	0.1
pyr1.oblique.tsem.branch.PDF.std	10	10	10	10

all_dendrites = Basal dendrites, pyr1.tuft = Tuft dendrites, pyr1.oblique = Oblique dendrites, B_inf = Branching rate, E = Branching constraint, eri.PDF.mean = Elongation rate mean, eri.PDF.std = Elongation rate stdev, tsem.branch.PDF.mean = Initial segment length mean, tsem.branch.PDF.std = Initial segment length stdev.

search, used in RA to find approximate roots, was parallelized and Netmorph simulations were also run in parallel.

The model was optimized in different phases. Basal, tuft and oblique dendrites were optimized separately, first for branching parameters and then for elongation parameters. The optimization function was selected as the euclidean distance between the model output and the target values.

The optimization was performed by stochastic Nelder-Mead in combination with Retrospective Approximation. Nelder-Mead used stochastic extensions [5] and Dennis-Woods stopping criteria or a maximum of 10 iterations. Retrospective Approximation ran for 10 iterations, with the sample size doubling at each iteration.

3.4.5 Optimization evaluation

The Kolmogorov-Smirnov test was used to test for equality of the data distributions and Netmorph generated distributions.

3.4.6 Statistical difference testing

The experimental data distributions were tested for equality with ANOVA. To test for equality of variances, Levene's test was used with median values. In the case of a difference each pair of distributions were tested separately with Student's t-test with Bonferroni corrections for multiple testing.

4 Results

The data was divided into groups by cortex (Barrel cortex - BC, Visual cortex - VC) and by neuronal population (etv, glt). The data was first tested for environmental factors that might explain some variation in data (4.1), and for groupwise differences with ANOVA (4.2). A Netmorph model was then constructed for each data group and the model parameters were optimized to fit the model with data (4.3). The optimization was evaluated with the Kolmogorov-Smirnov test to see if the optimized model distribution was equal to the data distribution (4.4). Finally, the model was used to generate some neurons and a large-scale network by using the optimized model parameters (4.5).

4.1 Environmental factor correlation

Known environmental factors (age, brain size) were tested for correlation with statistical properties. Age and brain size were tested for correlation with the number of branches, the intermediate segment lengths and the terminal segment lengths. The correlation results are shown in Table 4. The full plots of factors vs. data can be seen in Appendix A.

Table 4: Age and brain size related correlation, correlation coefficient and p-value.

AGE

	BC etv		BC glt		VC etv		VC glt	
Value	$\Delta y/\Delta x$	p-value	$\Delta y/\Delta x$	p-value	$\Delta y/\Delta x$	p-value	$\Delta y/\Delta x$	p-value
Basal - endings	-	0.2101	-	0.5339	-	0.3844	-	0.4960
Basal - IS length	-	0.3878	-	0.7285	-	0.4106	-	0.9560
Basal - TS length	1.6185	0.0416	0.9343	0.0428	-	0.3206	-	0.8017
Oblique - endings	-	0.7555	-	0.5865	-0.4995	0.0082	-	0.6136
Oblique - IS length	-	0.7449	-	0.7179	-	0.7073	-	0.7766
Oblique - TS length	-	0.1629	-	0.8502	-	0.7792	-	0.8773
Tuft - endings	-0.5827	0.0370	-	0.0621	-	0.0595	-	0.1652
Tuft - IS length	-	0.7909	-	0.6617	-	0.4591	-	0.5943
Tuft - TS length	-	0.7690	-	0.1511	-	0.2837	-	0.1979

BRAIN SIZE

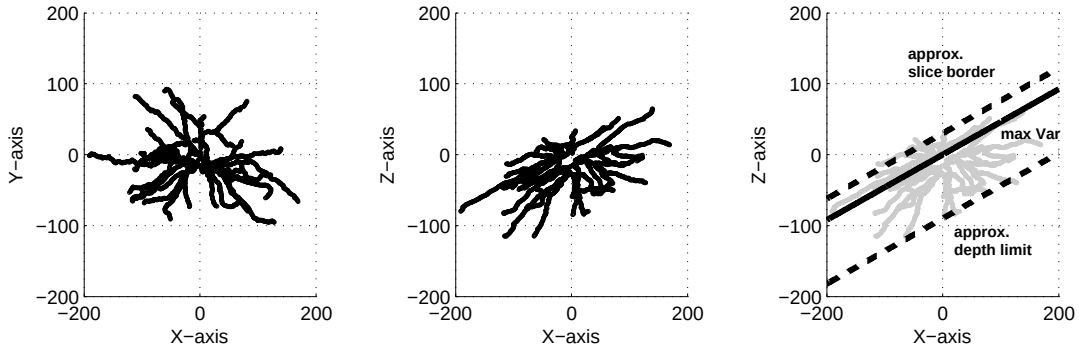
	BC etv		BC glt		VC etv		VC glt	
Value	$\Delta y/\Delta x$	p-value	$\Delta y/\Delta x$	p-value	$\Delta y/\Delta x$	p-value	$\Delta y/\Delta x$	p-value
Basal - endings	-	0.5083	-0.0054	0.0228	-	0.3481	-	0.1982
Basal - IS length	-	0.9837	-	0.4520	-	0.8160	-	0.8535
Basal - TS length	0.0350	0.0231	-	0.4980	-	0.1861	-	0.4905
Oblique - endings	-	0.8154	-	0.5234	-	0.9294	-	0.6933
Oblique - IS length	-	0.9851	-	0.4536	-	0.4361	-	0.0726
Oblique - TS length	-	0.3520	-	0.4766	-	0.0539	-	0.3067
Tuft - endings	-	0.4657	-	0.3119	-	0.4680	-	0.5552
Tuft - IS length	-	0.6232	-	0.1203	-	0.7403	-	0.8939
Tuft - TS length	-	0.9529	-	0.6420	-	0.9232	-	0.1356

BC = barrel cortex, VC = visual cortex, $\Delta y/\Delta x$ = linear regression coefficient, Basal = basal dendrites, Oblique = oblique dendrites, Tuft = tuft dendrites, Endings = number of endings in dendrite, TS length = terminal segment lengths, IS length = intermediate segment lengths.

4.1.1 Histological artifacts

Tissue compression was studied by plotting the coordinates of basal dendrite nodes in the XY and XZ planes. These are shown in Figure 1(a)-(b). The dendrite growth was expected to be the same in all directions. Flattened formation is apparent in the XZ plane, which was assumed to be caused by compression.

To study dendrite cutting the location of the slice border needs to be estimated. The amount of compression in the (slice) Z-direction was studied by Egger et al [14] to be about 45%, which is in accordance with empirical observation of about 50% by Dr. Krieger. If the original dendrite growth is assumed equal in all directions, the direction with most variation would then run parallel to the slice border. The border should be some 20-30μm up from the cell body (empirical observation), and the depth-view of the microscope extends to approximately 90μm down (empirical observation). The approximated borders are shown in Figure 1(c).



(a) Coordinates in XY-plane (b) Coordinates in XZ-plane (c) Approximate slice borders

Figure 1: Basal dendrite coordinates in XY and XZ-planes for BC-ETV neuron.

The dendrite ending location was computed as a distance from the approximated slice upper border line. The dendrite length and diameter are shown as a function of ending distance from slice border in Figure 2.

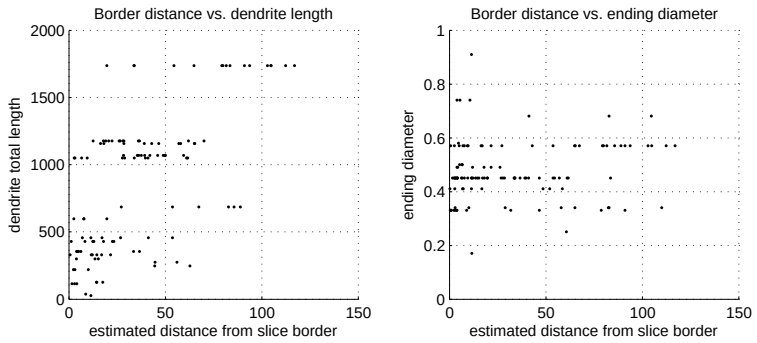


Figure 2: Dendrite length and diameter by distance from approximate slice border.

Tissue compression artifacts were not considered for correction as spatial statis-

tics were not used in the analysis of neurons. Dendrite cutting might occur and be affecting the current analysis. Correction would require further research and was left out from the present study.

4.2 Groupwise difference testing with ANOVA

Experimental data was tested for groupwise differences by ANOVA. To test for equal variance between groups Levene's test was performed with medians. The results can be seen in Table 5.

Table 5: Testing for differences between groups in original data. Levene's test for equality of variances, one-way ANOVA and pairwise t-tests with Bonferroni correction.

			Pairwise t-test / P-value			
	Levene's	ANOVA	BC: etv/glt	VC: etv/glt	etv: BC/VC	glt: BC/VC
Basal - endings	0.002	–	$< 0.001^*$	1.000	0.046^*	1.000
Oblique - endings	< 0.001	–	0.020^*	0.736	0.011^*	0.046^*
Tuft - endings	0.681	< 0.001	$< 0.001^*$	0.764	1.000	0.002^*
Basal - IS length	0.510	0.021	0.030^*	1.000	1.000	0.040^*
Oblique - IS length	0.492	0.631	–	–	–	–
Tuft - IS length	< 0.001	–	$< 0.001^*$	0.151	0.071	$< 0.001^*$
Basal - TS length	0.010	–	0.018^*	0.398	1.000	1.000
Oblique - TS length	0.345	0.007	1.000	1.000	0.042^*	0.332
Tuft - TS length	< 0.001	–	$< 0.001^*$	$< 0.001^*$	1.000	1.000

Levene's = Levene's test for equality of variances, ANOVA = one-way ANOVA for equality of means in a group, BC = barrel cortex, VC = visual cortex, Basal = basal dendrites, Oblique = oblique dendrites, Tuft = tuft dendrites, Endings = number of endings in dendrite, TS length = terminal segment lengths, IS length = intermediate segment lengths.

4.3 Model optimization

All Netmorph models were optimized in each group by using the Retrospective Approximation algorithm with a Nelder-Mead direct search. The optimized parameters for models are shown in Table 6.

The optimized model generated distribution parameters are shown in Table 8 in Appendix C, along with the original data values.

4.4 Evaluating optimization

The distributions for the number of endings, intermediate- and terminal segment lengths were computed. A Kolmogorov-Smirnov test was used to test for equality between the model generated and data distributions. The results can be seen in Table 7 and in Appendix B.

Table 6: Optimized model parameters.

		B	E	ERI [$\mu m/day$]	TSEM [μm]
BC-etv	basal	1.83	0.14	7.22 ± 7.90	0.43 ± 8.59
	tuft	5.94	0.32	103 ± 8.84	0.83 ± 16.67
	oblique	1.07	0.04	8.93 ± 9.92	0.41 ± 8.27
BC-glt	basal	1.72	0.42	10.8 ± 3.90	6.49 ± 7.71
	tuft	6.0	0.25	4.15 ± 21.0	1.03 ± 54.8
	oblique	1.24	0.32	13.8 ± 7.70	4.95 ± 5.10
VC-etv	basal	1.87	0.37	7.15 ± 7.83	0.43 ± 8.51
	tuft	6.00	0.33	18.0 ± 2.03	1.76 ± 0.58
	oblique	1.10	0.32	14.4 ± 7.17	0.05 ± 17.9
VC-glt	basal	2.13	0.51	10.9 ± 5.83	10.8 ± 0.26
	tuft	5.92	0.31	24.5 ± 12.5	5.46 ± 14.0
	oblique	1.34	0.62	13.1 ± 7.31	10.4 ± 0.36

B = branching probability parameter, E = branch constraining parameter, ERI = segment elongation rate distribution, TSEM = segment initial length distribution, BC = barrel cortex, VC = visual cortex, Basal = basal dendrites, Oblique = oblique dendrites, Tuft = tuft dendrites.

Table 7: Kolmogorov-Smirnov tests between experimental data and generated neurons.

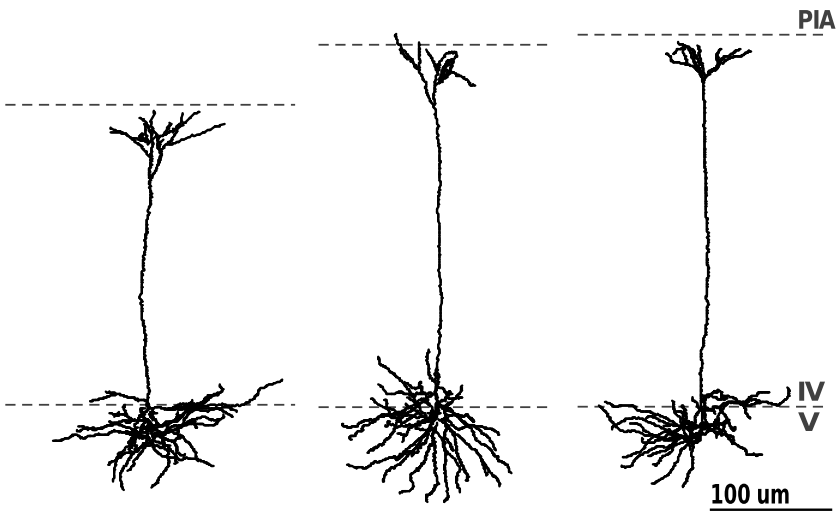
		Values from same dist.?			P-value			K-value		
		Endings	TS Len	IS Len	Endings	TS Len	IS Len	Endings	TS Len	IS Len
BC-etv	basal	Yes	No	No	0.974	0.000	0.000	0.038	0.193	0.089
	oblique	Yes	No	Yes	0.267	0.000	0.104	0.110	0.232	0.105
	tuft	Yes	No	No	0.685	0.000	0.000	0.147	0.878	0.582
BC-glt	basal	Yes	No	No	0.820	0.000	0.007	0.056	0.138	0.097
	oblique	Yes	Yes	No	0.866	0.155	0.000	0.056	0.076	0.248
	tuft	Yes	Yes	Yes	0.234	0.229	0.230	0.250	0.057	0.060
VC-etv	basal	Yes	No	No	1.000	0.000	0.002	0.027	0.171	0.098
	oblique	Yes	Yes	Yes	1.000	0.375	0.500	0.039	0.087	0.116
	tuft	Yes	No	No	0.409	0.044	0.000	0.210	0.097	0.187
VC-glt	basal	Yes	No	No	0.489	0.003	0.000	0.086	0.095	0.380
	oblique	Yes	Yes	No	0.991	0.782	0.000	0.046	0.055	0.308
	tuft	Yes	Yes	Yes	0.581	0.291	0.069	0.207	0.072	0.102

BC = barrel cortex, VC = visual cortex, Basal = basal dendrites, Oblique = oblique dendrites, Tuft = tuft dendrites, Endings = number of endings in dendrite, TS len = terminal segment lengths, IS len = intermediate segment lengths.

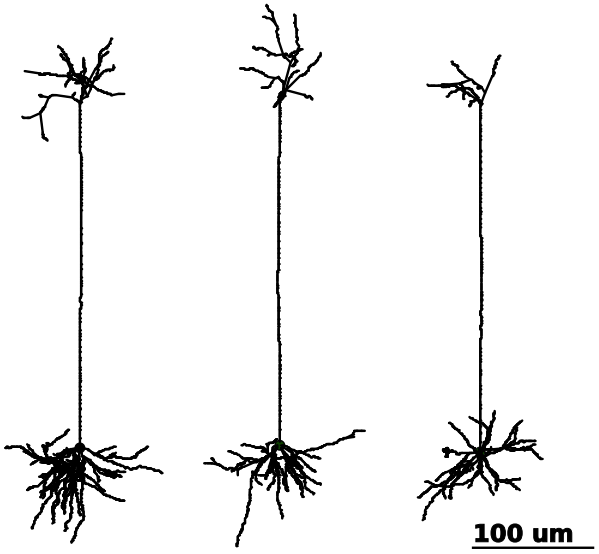
4.5 Neuron generation

The optimized model was used to generate neurons from each group. The generated neurons were visualized and compared to real neurons. These can be seen in Figure 3.

A large-scale network was generated from each group. Partial visualization of a network can be seen in Figure 4.



(a) Mouse barrel cortex, etv-population.



(b) Netmorph generated neurons.

Figure 3: Neurons from mouse barrel cortex and corresponding computationally generated neurons.

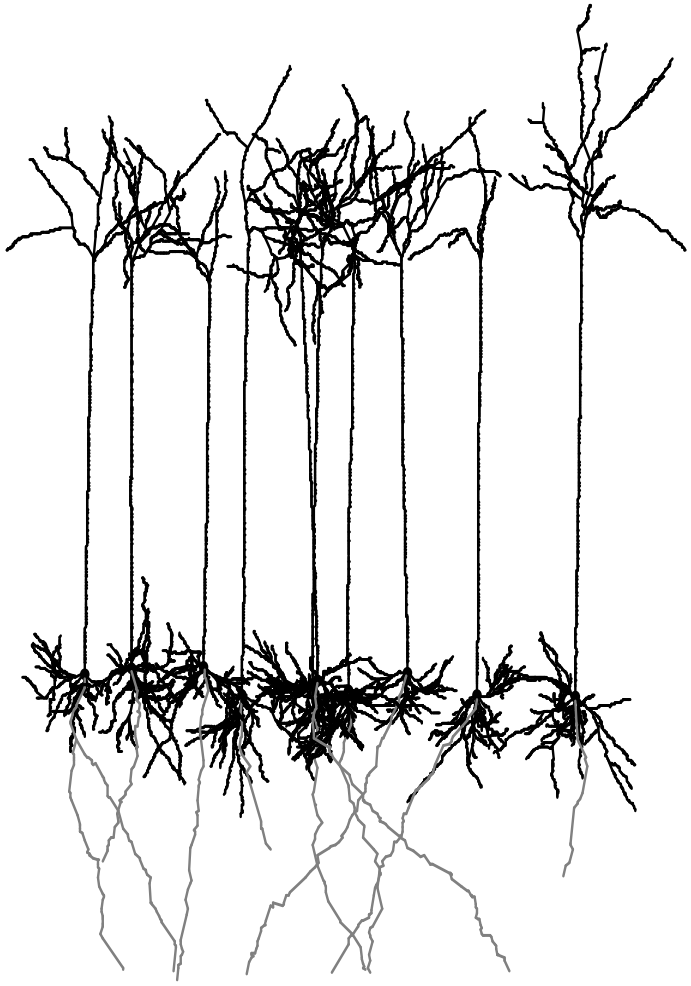


Figure 4: Netmorph generated network of neurons.

5 Discussion

The results presented in the previous chapter are discussed here. First environmental factors and histological artifacts are discussed in Sections 5.1-5.2. The groupwise differences in data are discussed in Section 5.3 and model optimization is discussed in Section 5.4.

5.1 Environmental factors

Environmental factors were tested to see if age or brain size affects the statistical properties of the neurons. The found correlations are discussed below:

Age - barrel cortex - basal dendrites - terminal segment lengths The barrel cortex basal dendrites had increased terminal segment lengths (p-values 0.0416 and 0.0428). Age varied from 18 to 25 days in the barrel cortex etv-population and 16 to 39 days in the barrel cortex glt-pyramids. The differences in terminal segment lengths between the youngest and oldest sample would be $11.3\mu m$ for bc-etv and $21.5\mu m$ for bc-glt. This could be possible neuroanatomically.

Age - visual cortex, etv - oblique dendrites - branches The visual cortex etv-population oblique dendrites had negative correlation between number of branches and age. The age span of the sample is narrow (22 to 25 days) and the retraction of branches does not make neuroanatomical sense, casting doubt on the significance of this correlation.

Age - barrel cortex, etv - tuft - branches The barrel cortex etv-population had negative correlation between age and number of branches in tuft. The result seems to be mostly because of a few young (P18) neurons with lots of endings.

Brain size - barrel cortex, etv - basal dendrites - terminal segment lengths The barrel cortex etv-population had an increased terminal segment length with increasing brain size. With neocortical thickness varying from $1081\mu m$ to $1582\mu m$, the difference in average length would be $17.5\mu m$ between the thinnest and thickest sample. This might be neuroanatomically plausible, as some research suggests [7] that the neurons are somewhat sparsely distributed in larger brains.

Brain size - barrel cortex, glt - basal dendrites - branches The barrel cortex glt-pyramids had less branches in basal dendrites, with neocortical thickness varying from $1028\mu m$ to $1414\mu m$. The difference in average number of branches would be 2.1 between the thinnest and thickest. The neuroanatomical sense of this possible decrease is not clear.

5.2 Histological artifacts

Slice compression was visible in the X,Z-plane plot of coordinates. It was assumed to be causing only spatial artifacts and was therefore left out of further consideration. Dendrite cutting at the slice border might be occuring, causing some dendrite terminal segment lengths to be lower than in reality. Segment length might be compensated or cut dendrites could be corrected by cloning existing dendrites or by simulating growth. Correction methodology would require further research and was not done.

5.3 Groupwise differences

Groups (BC etv, BC glt, VC etv and VC glt) were tested for equality, first by testing variance with Levene's test. Variances were different for five categories (basal - branches, oblique - branches, tuft - int. len, basal - term len, tuft - term len), and equal for the rest. For categories with equal variances, one-way ANOVA was done to test for equality of means. Four categories had different means (tuft - branches, basal - int len, oblique - int len, oblique - term len), leaving only one category (oblique - int len) with an equal mean and variance between groups.

A pairwise t-test was performed with Bonferroni corrections to determine which groupwise differences were statistically significant. The differences are discussed below by categories:

Basal dendrite branching Basal dendrite branching was different in the barrel cortex etv- and glt-pyramids, and between etv populations in the barrel cortex and the visual cortex. The BC-etv population is higher in layer 5 than BC-glt, while in the visual cortex the etv- and glt-pyramids are more mixed. As there is no difference in basal branching in the glt-pyramids, or between visual cortex etv- and glt-pyramids, it would suggest that the difference in BC-etv/BC-glt and BC-etv/VC-etv is by the higher location of the BC-etv population.

Oblique dendrite branching Oblique dendrite branching was different in BC-etv/BC-glt, BC-etv/VC-etv and BC-glt/VC-glt. The BC-etv/BC-glt and BC-etv/VC-etv differences might be explained by the close proximity of BC-etv to the L4/L5 border, and therefore less space for BC-etv oblique dendrites to grow. BC-glt and VC-glt are located on average the same distance from the L4/L5 border, although the cortex was thicker in the barrel cortex. The reason for the difference in oblique dendrite branching in this case is not clear.

Tuft dendrite branching Tuft dendrite branching was different for BC-etv/BC-glt and BC-glt/VC-glt.

Basal dendrite intermediate segment length Basal intermediate segment lengths were different for BC-etv/BC-glt and BC-glt/VC-glt.

Oblique dendrite intermediate segment length Oblique dendrite intermediate segment lengths were equal for all groups.

Tuft intermediate segment length Tuft intermediate segment lengths were different for BC-etv/BC-glt and BC-glt/VC-glt.

Basal dendrite terminal segment length Basal dendrite terminal segment lengths were different for BC-etv/BC-glt. The etv-pyramids are closer to the L4/L5 border which might affect their growth.

Oblique dendrite terminal segment length Oblique dendrite terminal segment lengths were different for BC-etv/VC-etv. The etv-pyramids in the barrel cortex are closer to the L4/L5 border which might affect their growth.

Tuft terminal segment length Tuft terminal segment lengths were different for BC-etv/BC-glt and VC-etv/VC-glt, but not for BC-etv/VC-etv, or BC-glt/VC-glt. This might suggest a genetical population (etv, glt) caused factor.

5.4 Model optimization

The model parameters were optimized with stochastic optimization methods. The goodness-of-fit between the optimized model and the experimental data was tested with a Kolmogorov-Smirnov test. The branching distributions for all groups were accepted as the same by K-S. Some length distributions were also accepted, but not all. The length optimization might be improved by changing the optimization algorithm or by considering another optimization function instead of the mean-square difference. The distributions generated by the optimized model seem to be quite good nevertheless (see Appendix C), even if they are not accepted by K-S. The input data might also be noisy, for example by dendrite cutting at slice borders (see 4.1.1).

Of the original data groups, all dendrite categories (basal, oblique, tuft) had statistical differences in branching between cortical areas (barrel, visual) and/or by genetical population (etv, glt). As the Kolmogorov-Smirnov test accepted the experimental data distributions as the same as Netmorph generated distributions, it could be argued that the differences in Netmorph branching parameters for the corresponding groups are statistically significant. These parameters are discussed below.

Basal dendrites For basal dendrite branching, BC-etv/BC-glt and BC-etv/VC-etv were statistically different. The corresponding optimized parameters are (1.83, 0.14) / (1.72, 0.42) and (1.83, 0.14) / (1.87, 0.37). In both cases there is a large difference in the E parameter $0.14 \rightarrow 0.42, 0.14 \rightarrow 0.37, E \in [0, 1]$. The E parameter has been suggested to denote competition for resources between dendrite branches, meaning that BC-etv population basal dendrites would have more resources to grow, and would therefore branch more.

Oblique dendrites For oblique dendrite branching, all were statistically different except for vc-etv/vc-glt. The branching parameters are BC-etv (1.07, 0.04), BC-glt (1.24, 0.32), VC-etv (1.10, 0.32), VC-glt (1.34, 0.62). For BC

etv/glt and BC-etv/VC-etv, the difference in the E parameter suggests that the BC-etv pyramids have less restricted oblique dendrite branching (0.04 vs 0.32/0.32). In the case of BC-glt/VC-glt, the E parameter is even higher in VC-glt, suggesting that they have the most restricted branching (0.32 vs 0.62).

Tuft Tuft dendrite branching was different in BC-etv/BC-glt and in BC-glt/VC-glt. The parameters were BC-etv (5.94, 0.32), BC-glt (6.0, 0.25), VC-etv (6.00, 0.33) and VC-glt (5.92, 0.31). These are almost the same for all groups, with BC-glt having slightly lower E-parameter, meaning it would branch a little more in the tuft.

5.5 Neuron generation

After optimizing the model parameters, some neurons were generated succesfully. On visual comparison they seem to be similar to real neurons. A large-scale network was also generated and visualized. Further work could involve comparing the large-scale networks in different cortical areas and/or populations. Electrophysiological simulation of large-scale networks could also be performed with e.g. NEURON software.

6 Conclusion

The initial research objectives were defined as:

O1 Can we use computational algorithms to simulate growth of L5 pyramidal neurons?

O2 Are the generated neurons statistically similar to experimental data?

O3 Can we create a large scale network from the generated neurons and simulate neuronal activity?

It was demonstrated that the Netmorph framework can be used to simulate growth of realistic morphologies of L5 pyramidal neurons (in the case of mouse neocortex). The generated neurons were at least partially statistically similar to experimental data. Various environmental factors were considered which might affect the statistics of the experimental data, and which are not present in simulation. A large-scale network was also constructed, and it was possible to use it for neuronal activity simulation with NEURON or some other electrophysiological simulator.

Future work could involve comparison of a large-scale network between cortical areas, and electrophysiological simulation of these networks. The cases where model optimization did not produce an exactly corresponding distribution for some variables (e.g. segment lengths) could be studied more to see if it was caused by external factors or noise in the original data, or if it is problem within the underlying Netmorph model.

References

[1] Ascoli, G.A. Progress and Perspectives in Computational Neuroanatomy. *The Anatomical Record* 1999, vol. 257, pp.195-207.

[2] Ascoli, G.A., Krichmar, J.L. L-neuron: A modeling tool for the efficient generation and parsimonious description of dendritic morphology. *Neurocomputing* 2000, vol. 32-33, pp.1003-1011.

[3] Ascoli, G.A., Krichmar, J.L., Nasuto, S.J., Senft, S.L. Generation, description and storage of dendritic morphology data. *Phil. Trans. R. Soc. Lond. B* 2001, vol. 356, pp.1131-1145.

[4] Ascoli, G.A. (ed.). *Computational Neuroanatomy*. USA, Humana Press, 2002.

[5] Barton R.R., Ivey, J.S. Jr. Nelder-Mead Simplex Modifications for Simulation Optimization. *Management Science* Jul 1996, vol. 42(7), pp.954-973.

[6] Bonferroni, C.E. Il calcolo delle assicurazioni su gruppi di teste. In: *Studi in Onore del Professore Salvatore Ortu Carboni*. Rome, 1935.

[7] Braitenberg, V., Schüz, A. *Cortex: Statistics and geometry of neuronal connectivity* 2nd edition. Berlin, Springer-Verlag, 1998.

[8] Brown, M.B., Forsythe, A.B. Robust tests for equality of variances. *Journal of the American Statistical Association* 1974, vol. 69, pp.364-367.

[9] Broyden, C.G. The convergence of a class of double-rank minimization algorithms. *IMA Journal of Applied Mathematics* 1969, vol. 6(1), pp.76-90.

[10] Burke, R.E., Marks, W.B., Ulfhake, B. A parsimonious description of motoneuron dendritic morphology using computer simulation. *The Journal of Neuroscience* 1992 Jun, vol. 12(6), pp.2403-2416.

[11] Chen, H., Schmeiser, B.W. Stochastic root finding via retrospective approximation. *IIE Transactions* 2001, vol. 33, pp.259-275.

[12] Crossman, A.R., Neary, D, *Neuroanatomy* 3rd edition. Edinburgh, Churchill Livingstone, 2005.

[13] Donohue, D.E., Ascoli, G.A. Local diameter fully constrains dendritic size in basal but not apical trees of CA1 pyramidal neurons. *J. Comp. Neuroscience* 2005, vol. 19, pp.223-238.

[14] Egger V., Nevian, T., Bruno, R.M. Subcolumnar dendritic and axonal organization of spiny stellate and star pyramid neurons within a barrel in rat somatosensory cortex. *Cerebral cortex* 2008 Apr, vol. 18, pp.876-889.

[15] Ethier, S.N., Kurtz, T.G, *Markov Processes*. New Jersey, John Wiley & Sons, 1986.

[16] Fletcher, R. A new approach to variable metric algorithms. *The Computer Journal* 1970, vol. 13(3), pp.317-322.

[17] Fletcher, R, *Practical methods of optimization* 2nd edition. Great Britain, John Wiley & Sons, 1987.

[18] Fox, K, *Barrel Cortex*. Cambridge, Cambridge University Press, 2008.

[19] Glaser, J.R., Glaser, E.M. Neuron imaging with Neurolucida - A PC-based system for image combining microscopy. *Computerized Medical Imaging and Graphics* 1990, vol. 14(5), pp.307-317.

[20] Geisser, S., Greenhouse, S.W. An Extension of Box's Results on the Use of the F Distribution in Multivariate Analysis. *The Annals of Mathematical Statistics* Sep 1958, vol. 29(3), pp.885-891.

[21] Goldfarb, D. Factorized Variable Metric Methods for Unconstrained Optimization. *Mathematics of Computation* Oct 1976, vol. 30(136), pp.796-811.

[22] Groh, A., Meyer, H.S., Schmidt, E.F., Heintz, N., Sakmann, B., Krieger, P. Cell-type specific properties of pyramidal neurons in neocortex underlying a layout that is modifiable depending on the cortical area. *Cerebral cortex* 2010 Apr, vol. 20(4), pp.826-36.

[23] Harris, T.E, *The theory of branching processes*. Berlin, Springer-Verlag, 1963.

[24] Hillman, D.E. Neuronal shape parameters and substructures as a basis of neuronal form. In: *The neurosciences: Fourth study program*. USA, The MIT Press, 1979.

[25] Hu, J., Fu, M.C., Marcus, S.I. A model reference adaptive search for global optimization. *Operations Research* 2007, vol. 55(3), pp.549-568.

[26] Hu, J., Fu, M.C., Marcus, S.I. A model reference adaptive search method for stochastic global optimization. *Communications in information and systems* 2008, vol 8(3), pp.245-276.

[27] Huynh, H., Feldt, L.S. Estimation of the Box Correction for Degrees of Freedom from Sample Data in Randomized Block and Split-Plot Designs. *Journal of Educational Statistics* 1976, vol. 1(1), pp. 69-82.

[28] Jagers, P. Population-size-dependent branching processes. *Journal of Applied Mathematics and Stochastic Analysis* 1996, vol. 9(4), pp.449-457.

[29] Jagers, P., Klebaner, F.C. Population-size-dependent and age-dependent branching processes. *Stochastic Processes and their Applications* 2000, vol. 87, pp.235-254.

[30] Jungnickel, D. *Graphs, Networks and Algorithms* 3rd edition. Berlin, Springer-Verlag, 2008.

[31] Kimmel, M., Axelrod, D.E, *Branching processes in biology*. New York, Springer-Verlag, 2002.

[32] Koene, R.A., Tijms, B., van Hees, P., Postma, F., de Ridder, A., Ramakers, G.J.A., van Pelt, J., van Ooyen, A. NETMORPH: A framework for the stochastic generation of large scale neuronal networks with realistic neuron morphologies. *Neuroinform* 2009, vol. 7, pp.195-210.

[33] Kolmogorov, A. Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari* 1933, vol. 4, pp.83-91.

[34] Lehmann, E.L., Romano, J.P. *Testing Statistical Hypotheses* 3rd edition. USA, Springer Science+Business Media, 2005.

[35] Levene, H. Robust tests for equality of variances. In: *Contributions to Probability and Statistics*, Olkin I., Ghurye S.G., Hoeffding W., Madow W.G., Mann H.B. (eds.). USA, Stanford University Press, 1960.

[36] Massey, F.J. Jr. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association* Mar 1951, vol. 46(253), pp.68-78.

[37] Mauchly, J.W. Significance Test for Sphericity of a Normal n-Variate Distribution. *The Annals of Mathematical Statistics* Jun 1940, vol. 11(2), pp.204-209.

[38] Milliken, G.A., Dallas, E.J. *Analysis of messy data* 2nd edition. Boca Raton, Chapman & Hall / CRC, 2009.

[39] Møller, A.R, *Sensory systems*. USA, Elsevier Science, 2003.

[40] Nelder, J.A., Mead, R. A simplex method for function minimization. *The Computer Journal* 1965, vol. 7, pp.308-313.

[41] Nocedal, J., Wright, S.J. *Numerical Optimization* 2nd edition. USA, Springer Science+Business Media, 2006.

[42] Panico, J., Sterling, P. Retinal neurons and vessels are not fractal but space-filling. *The journal of comparative neurology* 1995, vol. 361, pp.479-490.

[43] Purves, D., Augustine, G.J., Fitzpatrick, D., Hall, W.C., LaMantia, A-S., McNamara, J.O., Williams, S.M. (eds.), *Neuroscience* 3rd edition. USA, Sinauer Associates, 2004.

[44] Quarteroni, A., Sacco, R., Saleri, F, *Numerical Mathematics* 2nd edition. Berlin, Springer, 2007.

[45] Rall W. Branching dendritic trees and motoneuron membrane resistivity. *Experimental Neurology* 1959, vol. 1(5), pp.491-527.

[46] Rall W. Theoretical significance of dendritic trees for neuronal input-output relations. In: *Neural theory and modeling*, Reiss R. (ed.). USA, Stanford University Press, 1964.

[47] Rall W. Time constants and electronic length of membrane cylinders and neurons. *Biophysical Journal* 1969, vol. 4, pp.1483-1508.

[48] Robbins, H., Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics* 1951 Sep, vol. 22(3), pp.400-407.

[49] Samsonovich, A.V., Ascoli, G.A. Statistical determinants of dendritic morphology in hippocampal pyramidal neurons: A Hidden Markov Model. *Hippocampus* 2005, vol. 15, pp.166-183.

[50] Schubert, D., Kötter, R., Luhmann, H.J., Staiger, J.F. Morphology, electrophysiology and functional input connectivity of pyramidal neurons characterizes a genuine layer Va in the primary somatosensory cortex. *Cerebral cortex* 2006 Feb, vol. 16, pp.223-236.

[51] Senft, S.L., Ascoli, G.A. Reconstruction of brain networks by algorithmic amplification of morphometry data. *Lecture Notes in Computer Science vol. 1606* pp.25-33. Berlin, Springer, 1999.

[52] Shanno, D.F. Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computation* Jul 1970, vol. 24(111), pp.647-656.

[53] Shepherd, G.M. (ed.), *The synaptic organization of the brain* 4th edition. New York, Oxford University Press, 1998.

[54] Sholl, D.A. Dendritic organization in the neurons of the visual and motor cortices of the cat. *Journal of Anatomy* 1953 Oct, vol. 87(4), pp.387-406.

[55] Shoukri, M.M., Pause, C.A., *Statistical methods for health sciences* 2nd edition. USA, CRC Press, 1999.

[56] Smirnov, N.V. On the estimation of discrepancy between empirical curves of distribution for two independent samples. *Moscow University Mathematics Bulletin* 1939.

[57] Spall, J.C. Stochastic optimization and the simultaneous perturbation method. *Proceedings of the 1999 Winter Simulation Conference* 1999, ISBN 0-7803-5780-9, pp.101-109.

[58] Spall, J.C. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on automatic control* 2000 Oct, vol. 45(10), pp.1839-1853.

[59] Spall, J.C, *Introduction to stochastic search and optimization*. New Jersey, John Wiley & Sons, 2003.

[60] Stuart, G., Spruston, N., Häusser, M. (eds.), *Dendrites*. Oxford, Oxford University Press, 1999.

[61] Tamori, Y. Theory of dendritic morphology. *Physical Review E* 1993 Oct, vol. 48(4), pp.3124-3129.

[62] Thas, O., *Comparing Distributions*. USA, Springer Science+Business Media, 2010.

[63] Tomick, J.J., Arnold, S.F., Barton, R.R. Sample size selection for improved Nelder-Mead performance. *Proceedings of the 1995 Winter Simulation Conference* 1995, pp.341-345.

[64] Uemura, E., Carriquiry, A., Kliemann, W., Goodwin, J. Mathematical modeling of dendritic growth in vitro. *Brain Research* 1995 Feb, vol. 671(2), pp.187-194.

[65] van Ooyen, A. (ed.), *Modeling Neural Development*. USA, The MIT Press, 2003.

[66] van Pelt, J., Verwer, R.W.H. The exact probabilities of branching patterns under terminal and segmental growth hypotheses. *Bulletin of Mathematical Biology* 1983, vol. 45(2), pp.269-285.

[67] van Pelt, J., Verwer, R.W.H. Topological properties of binary trees grown with order-dependent branching probabilities. *Bulletin of Mathematical Biology* 1986, vol. 48(2), pp.197-211.

[68] van Pelt, J., Uylings, H.B.M., Verwer, R.W.H., Pentney, R.J., Woldenverg, M.J. Tree asymmetry - a sensitive and practical measure for binary topological trees. *Bulletin of Mathematical Biology* 1992, vol. 54(5), pp.759-784.

[69] van Pelt, J., Dityatev, A.E., Uylings, H.B.M. Natural variability in the number of dendritic segments: model-based inferences about branching during neurite outgrowth. *The Journal of Comprarative Neurology* 1997, vol. 387, pp.325-340.

[70] van Pelt, J., van Ooyen, A., Uylings, H.B.M. Modeling dendritic geometry and the development of nerve connections. In: *Modeling in the Neurosciences, Reeke G.N. et al (eds.)*. Boca Raton USA, CRC-Press, 2001.

[71] van Pelt, J., Schierwagen, A. Morphological analysis and modeling of neuronal dendrites. *Mathematical Biosciences* 2004, vol. 188, pp.147-155.

[72] van Pelt, J., Uylings, H.B.M. Modeling Neuronal Growth and Shape. In: *Modeling Biology - Structures, Behaviors, Evolution; Laubichler, M.D. et al (eds.)* pp.195-215. Cambridge Massachusetts, The MIT Press, 2007.

[73] van Veen, M., van Pelt, J. A model for outgrowth of branching neurites. *J. Theor. Biol.* 1992, vol. 159, pp.1-23.

[74] van Veen, M.P., van Pelt, J. Terminal and intermediate segment lengths in neuronal trees with finite length. *Bulletin of Mathematical Biology* 1993, vol. 55(2), pp.277-294.

[75] Villacorta, J.A., Castro, J., Negredo, P., Avendaño, C. Mathematical foundations of the dendritic growth models. *J. Math. Biol* 2007, vol. 55, pp.817-859.

[76] Wang, Y., Gupta, A., Toledo-Rodriguez, M., Wu, C.Z., Markram, H. Anatomical, physiological, molecular and circuit properties of nest basket cells in the developing somatosensory cortex. *Cerebral cortex* 2002 Apr, vol. 12, pp.395-410.

[77] Washington, S.D., Ascoli, G.A., Krichmar, J.L. A statistical analysis of dendritic morphology's effect on neuron electrophysiology of CA3 pyramidal cells. *Neurocomputing* 2000, vol. 32-33, pp.261-269.

[78] Zubler, F., Douglas, R. A framework for modeling the growth and development of neurons and networks. *Frontiers in Computational Neuroscience* 2009 Nov, vol. 3, art. 25.

A Environmental factors

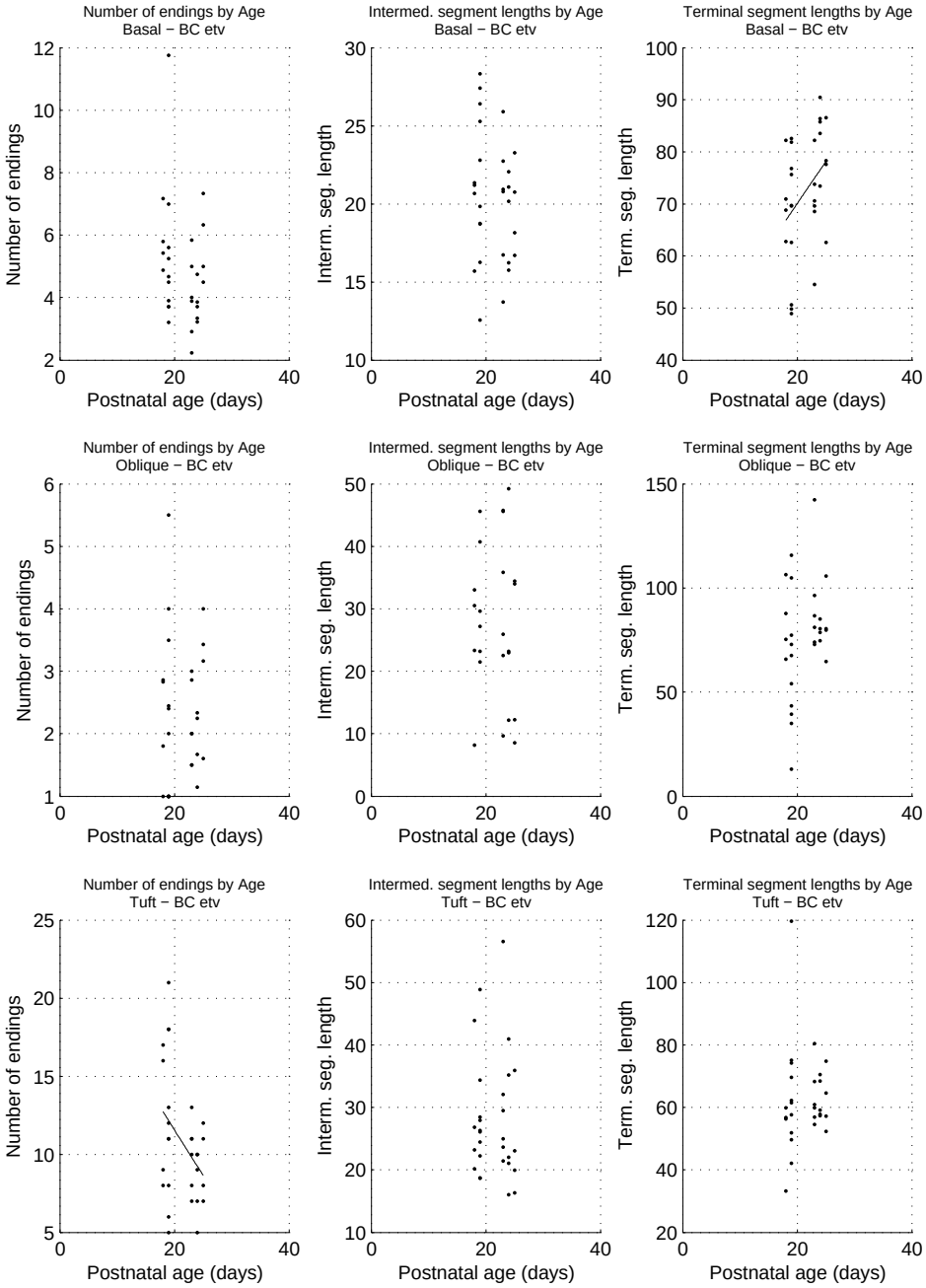


Figure 5: Dendrite measurements as a function of age, barrel cortex, etv

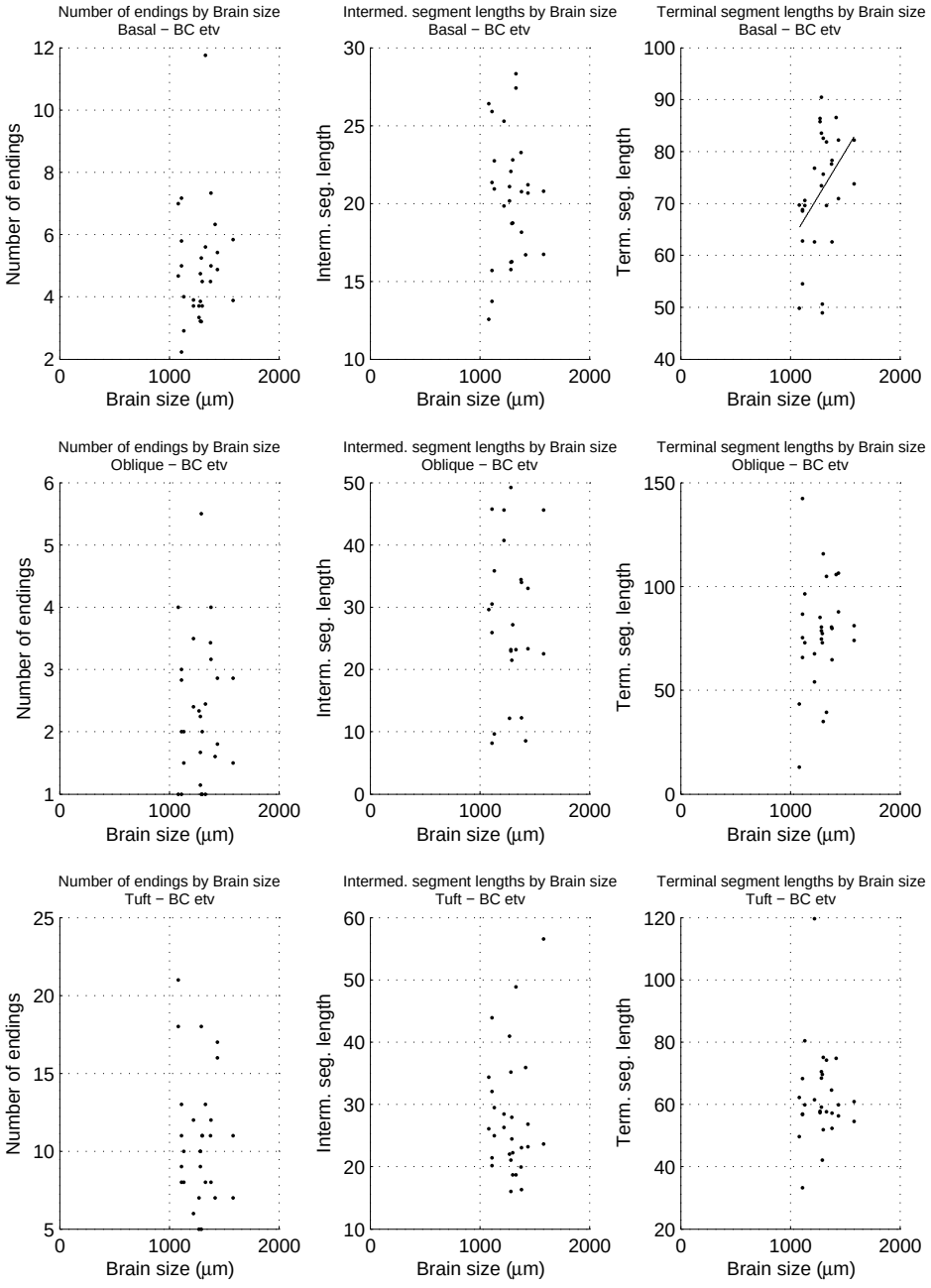


Figure 6: Dendrite measurements as a function of brain size, barrel cortex, etv

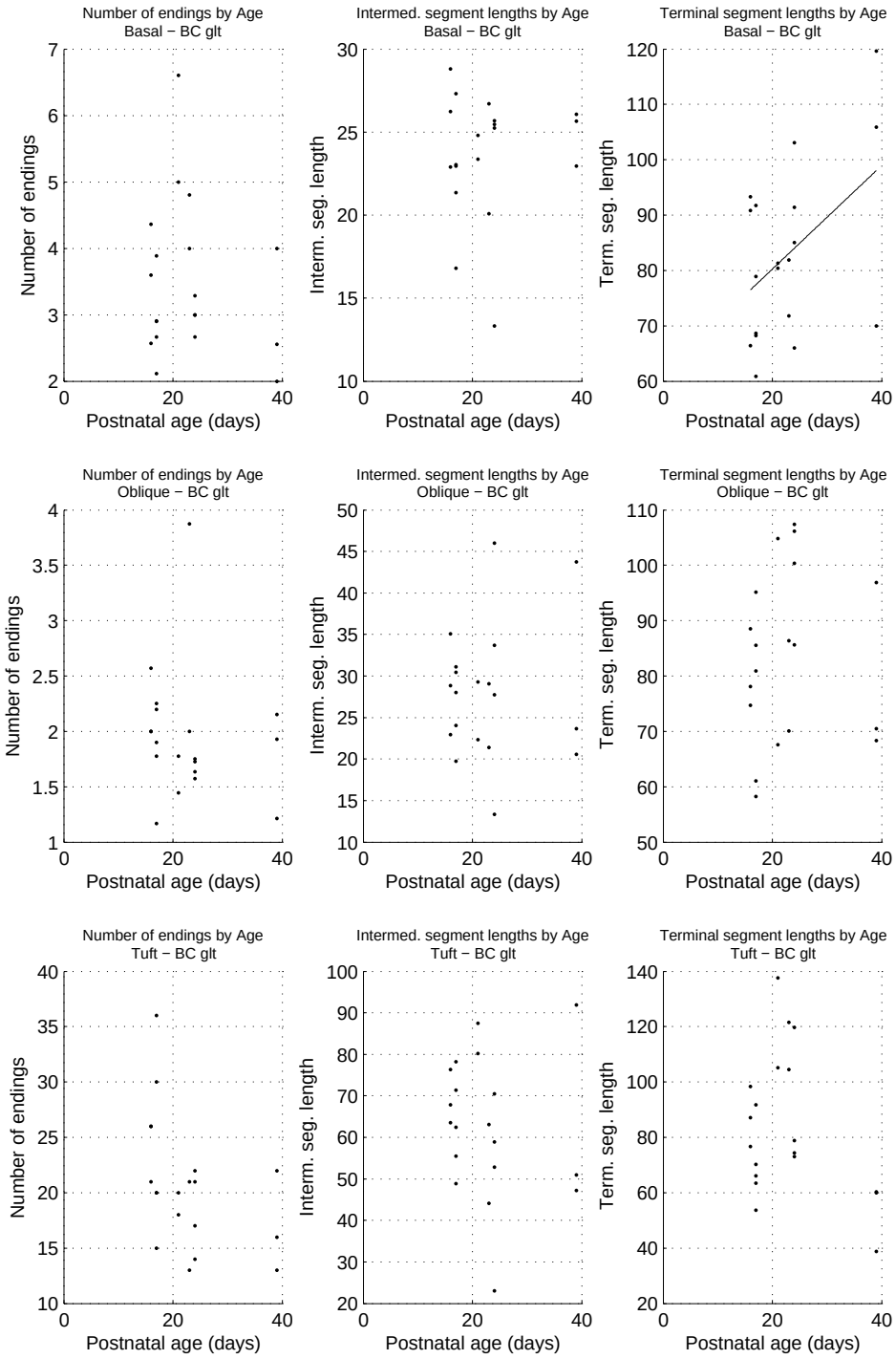


Figure 7: Dendrite measurements as a function of age, barrel cortex, glt

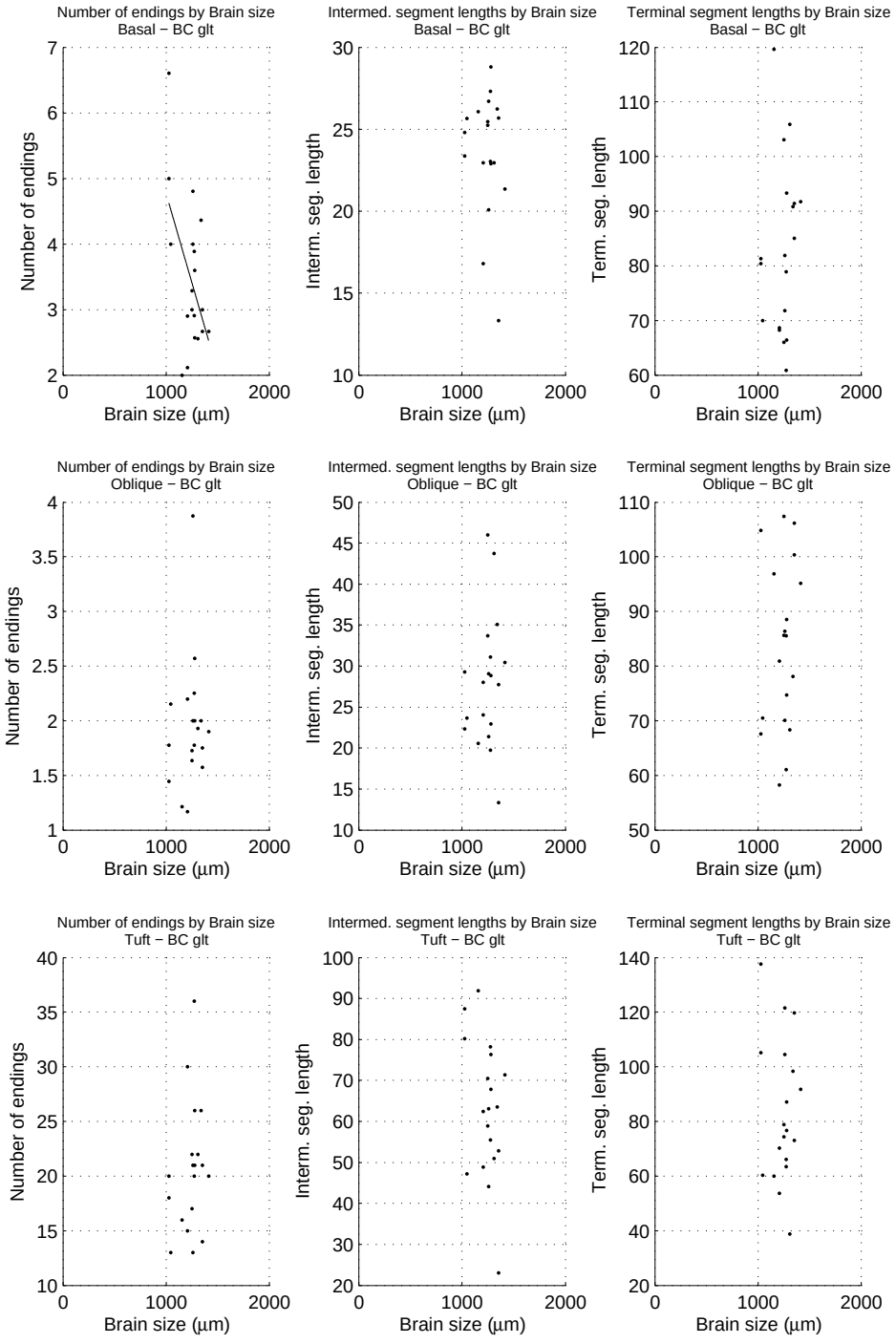


Figure 8: Dendrite measurements as a function of brain size, barrel cortex, glt

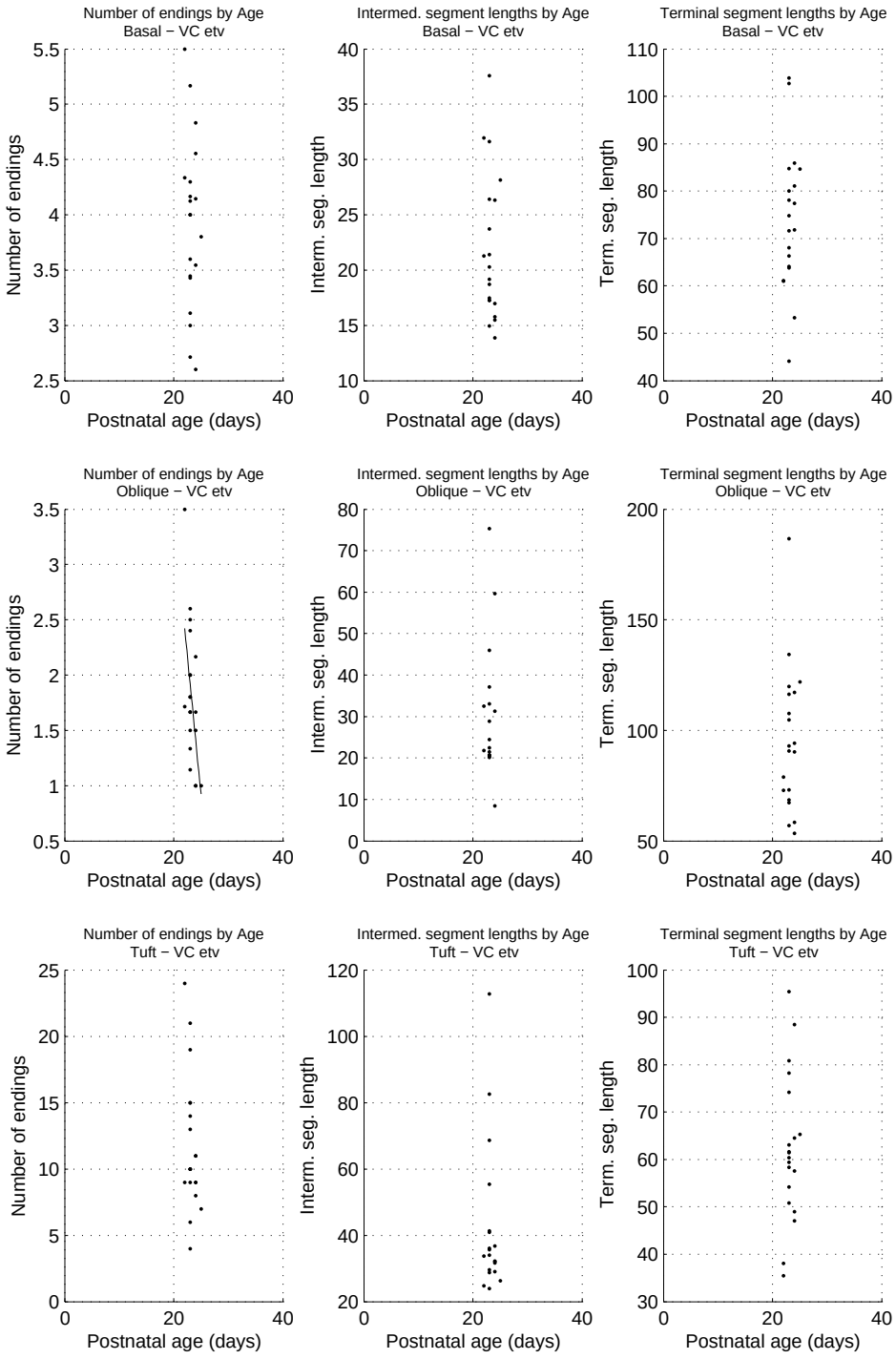


Figure 9: Dendrite measurements as a function of age, visual cortex, etv

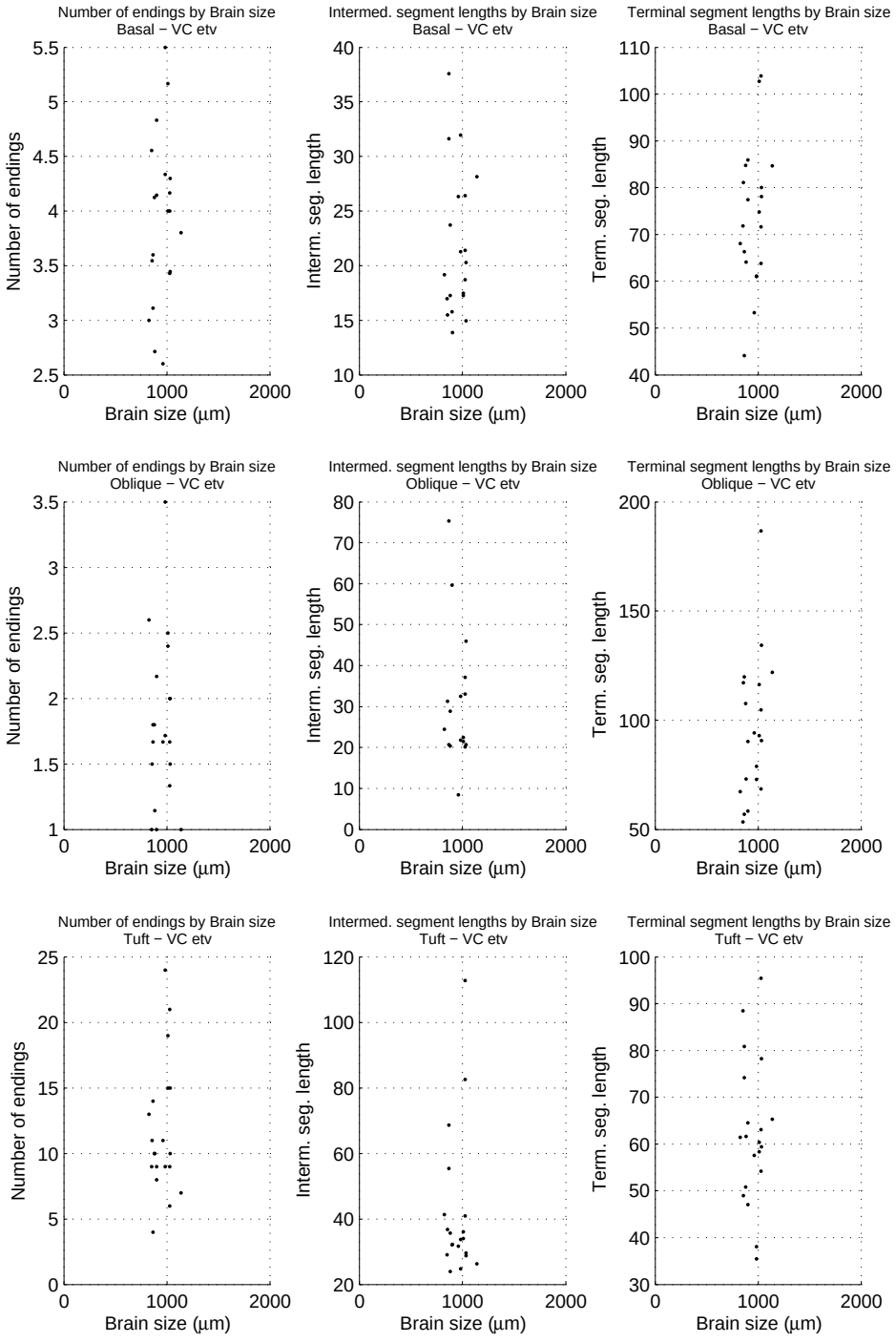


Figure 10: Dendrite measurements as a function of brain size, visual cortex, etv

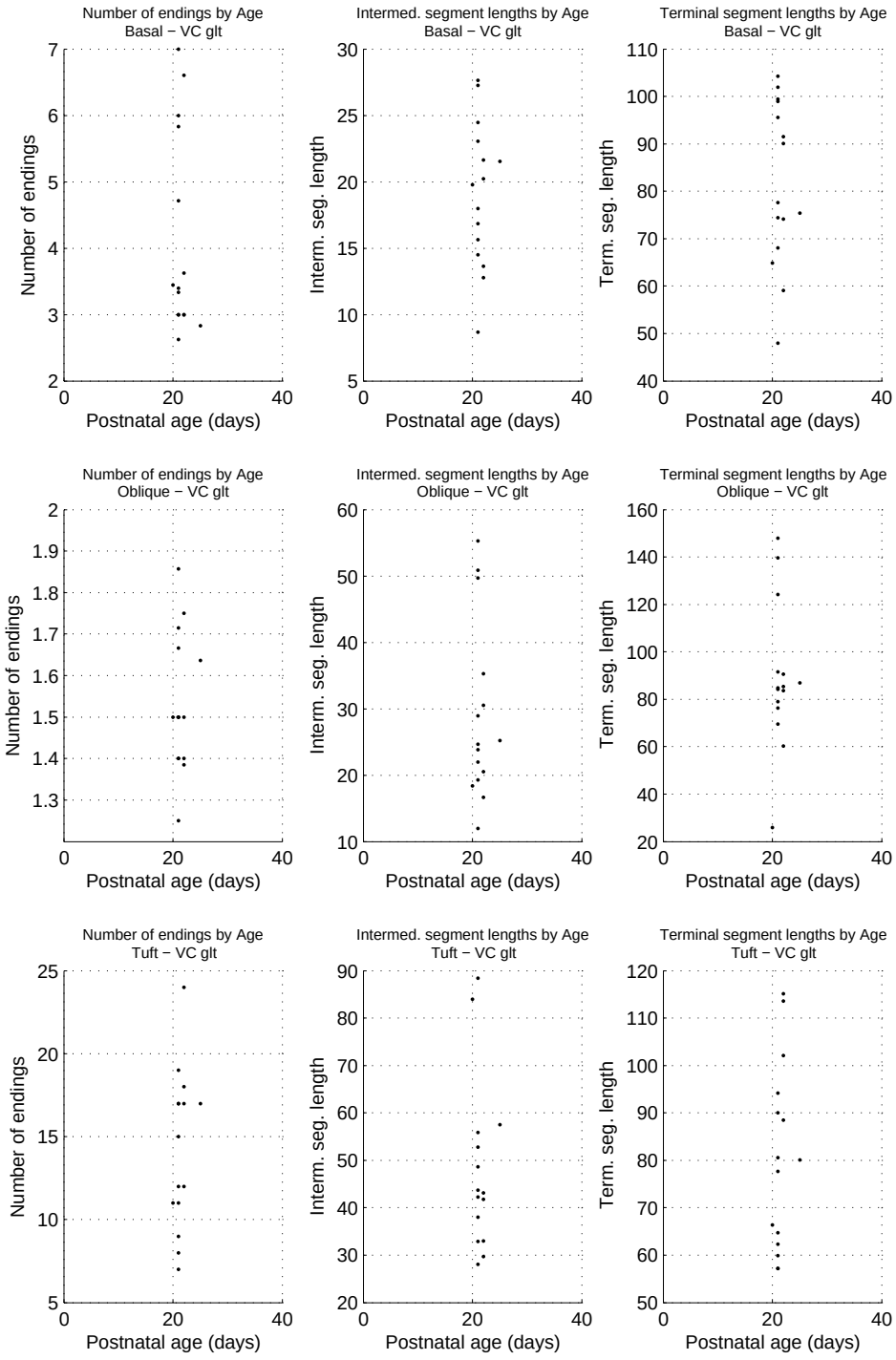


Figure 11: Dendrite measurements as a function of age, visual cortex, glt

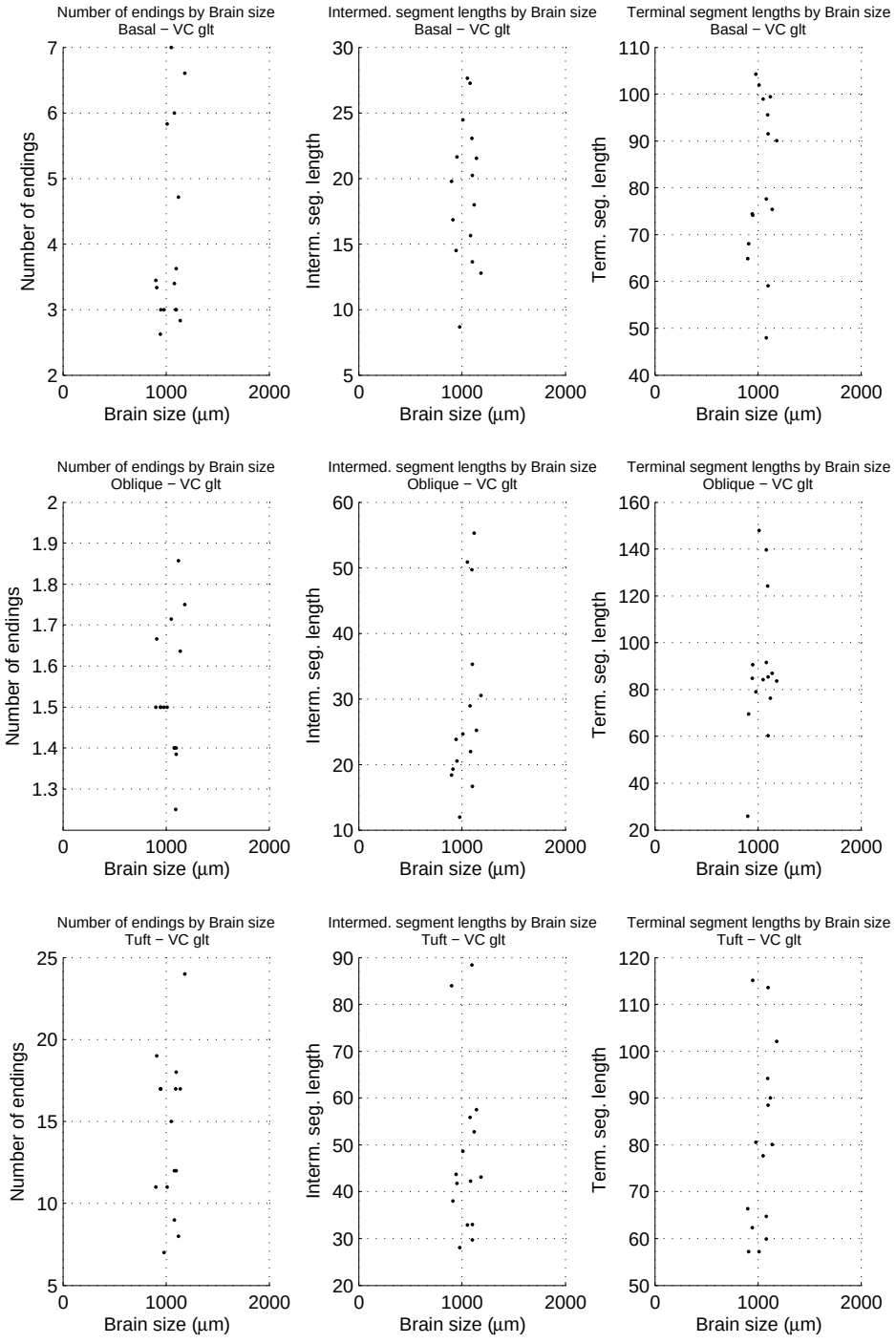


Figure 12: Dendrite measurements as a function of brain size, visual cortex, glt

B Segment distributions and Kolmogorov-Smirnov tests

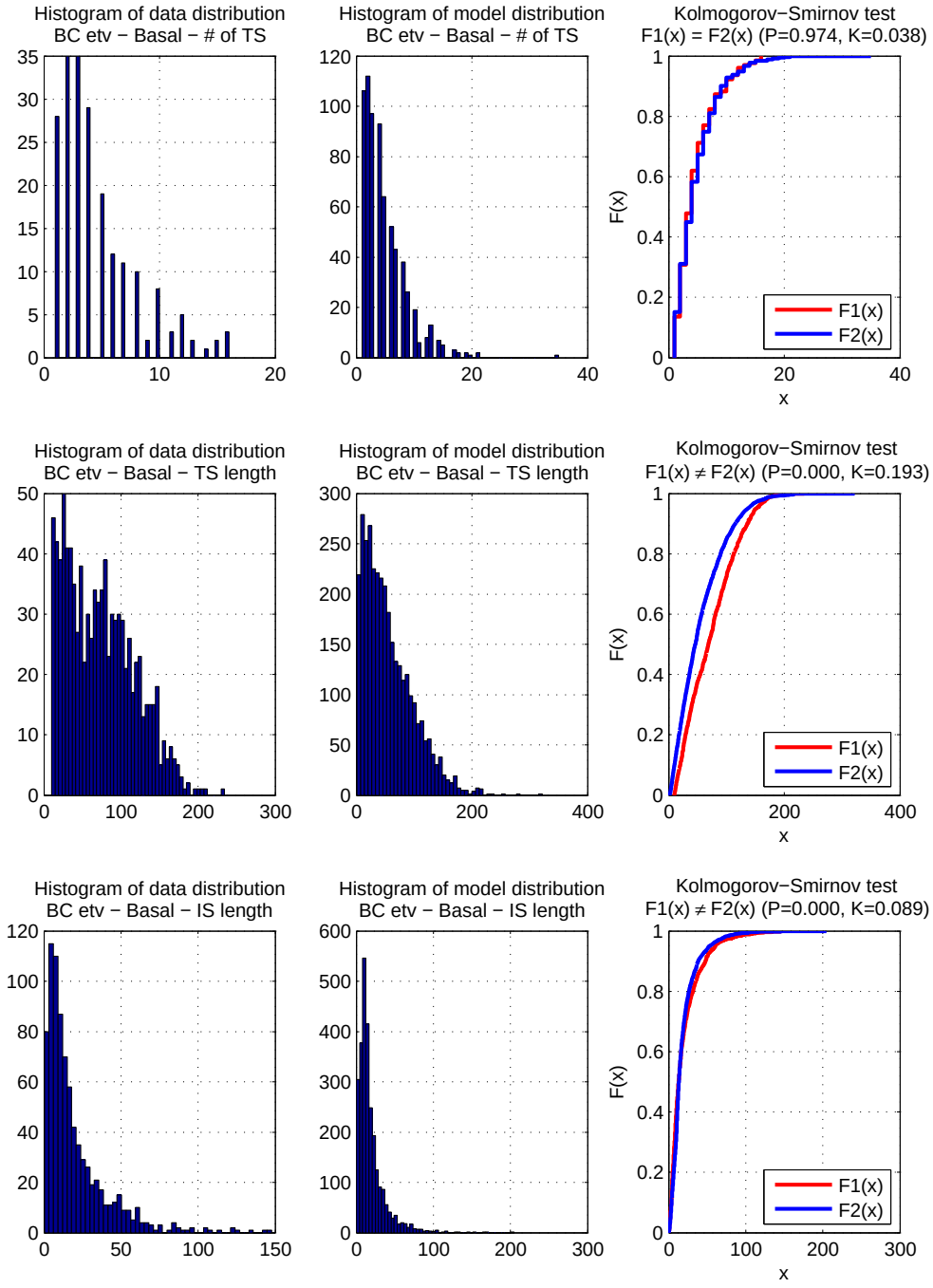


Figure 13: Barrel cortex, etv, basal dendrites

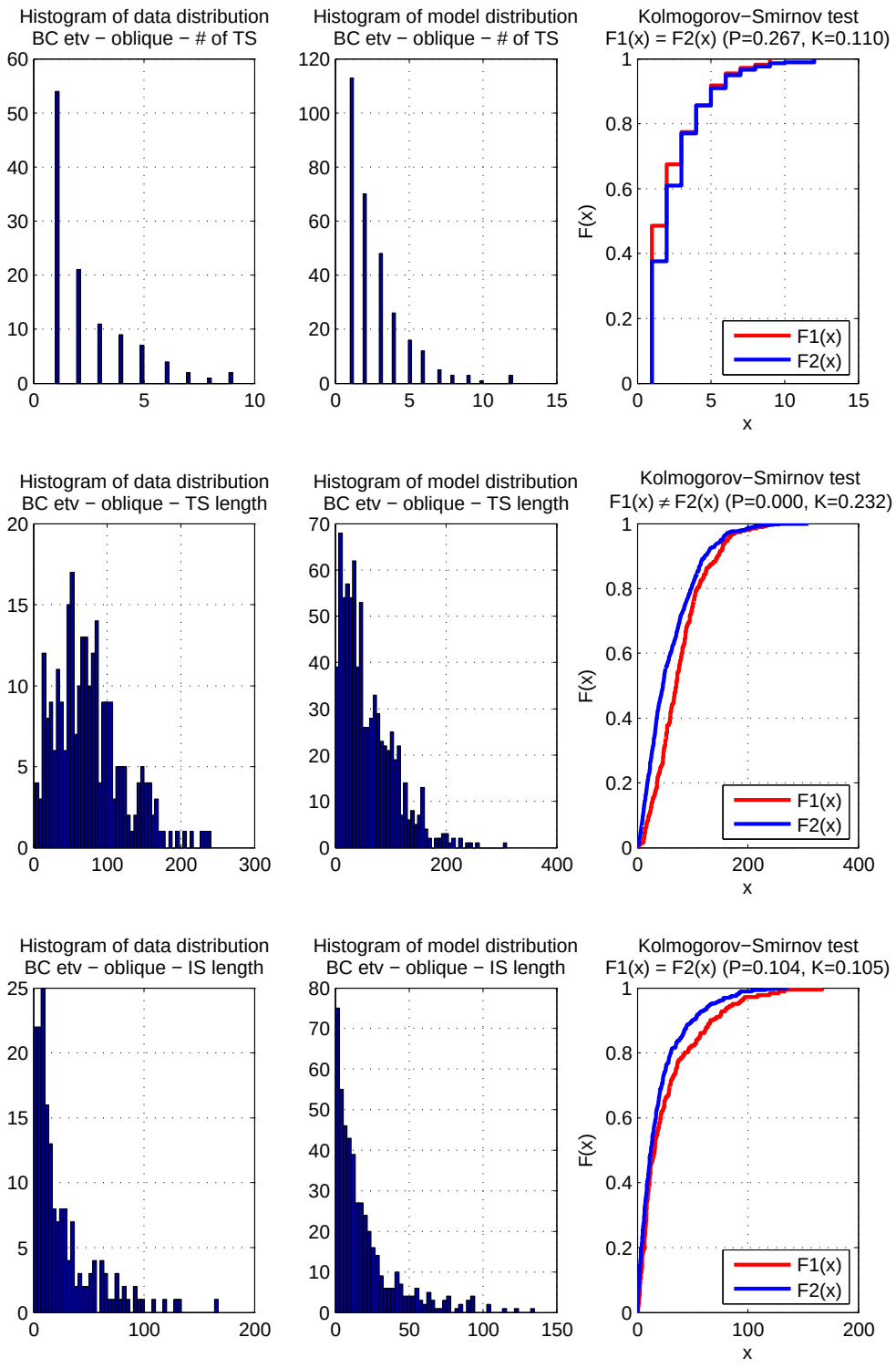


Figure 14: Barrel cortex, etv, oblique dendrites

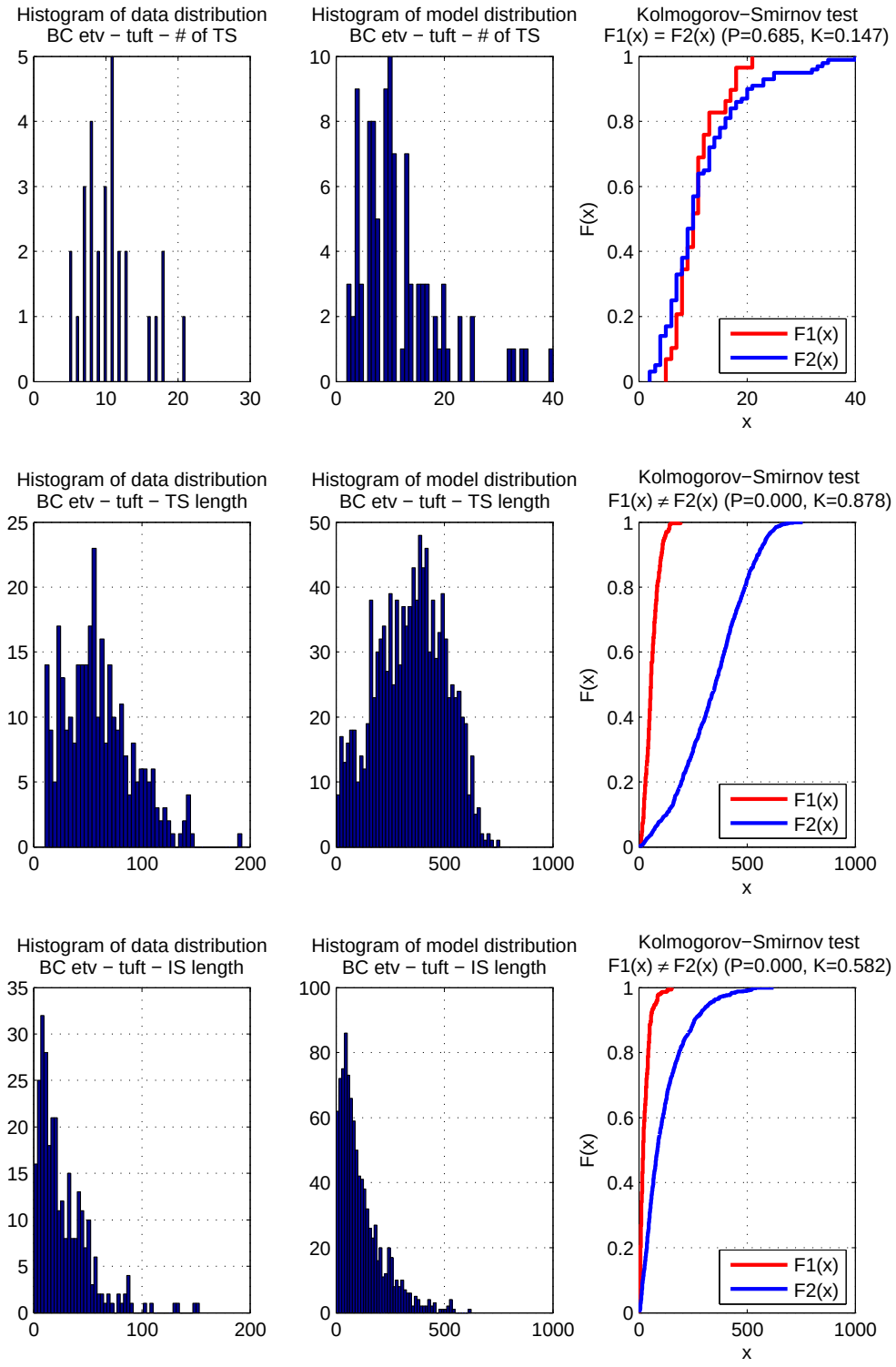


Figure 15: Barrel cortex, etv, tuft dendrites

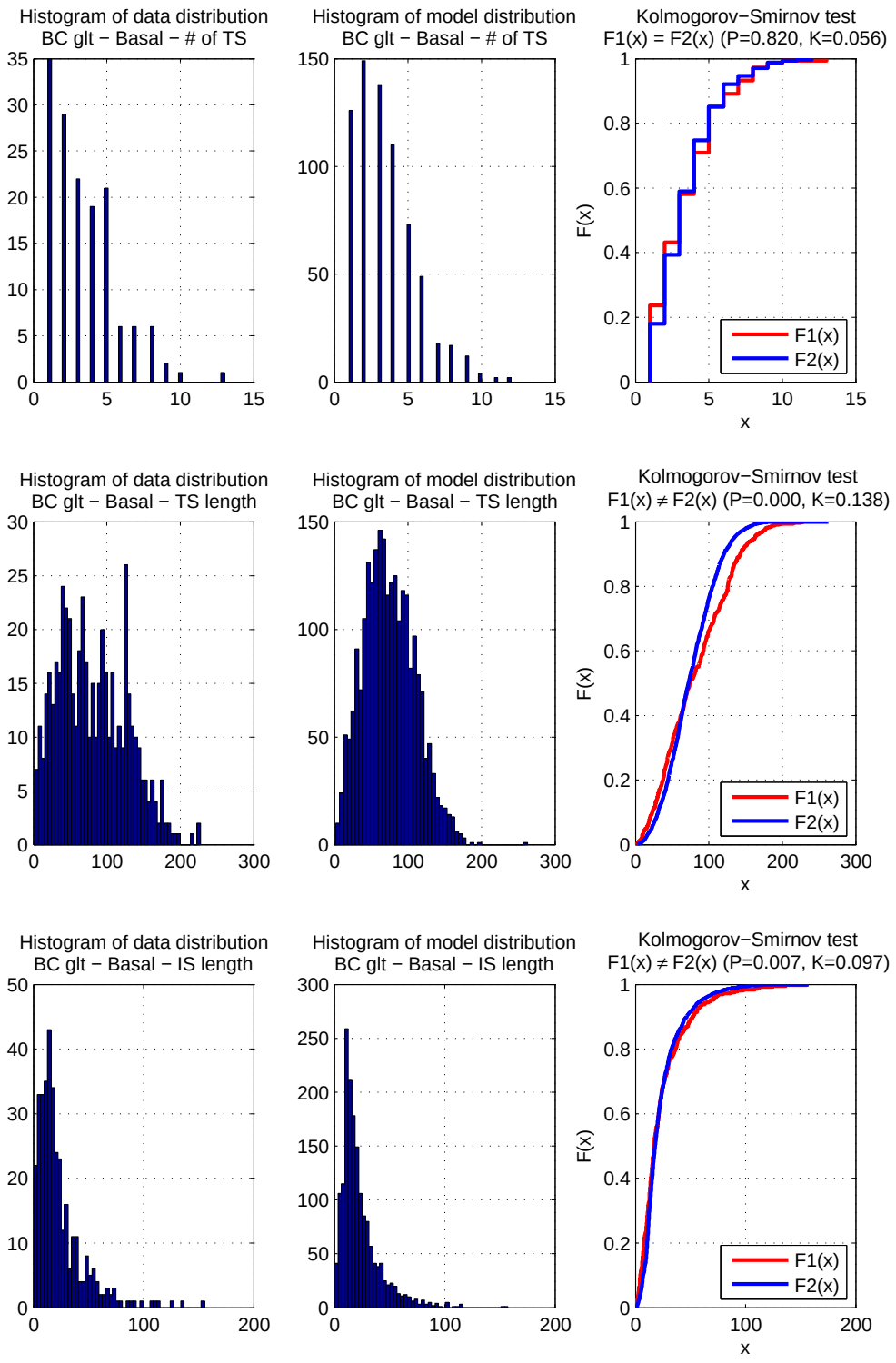


Figure 16: Barrel cortex, glt, basal dendrites

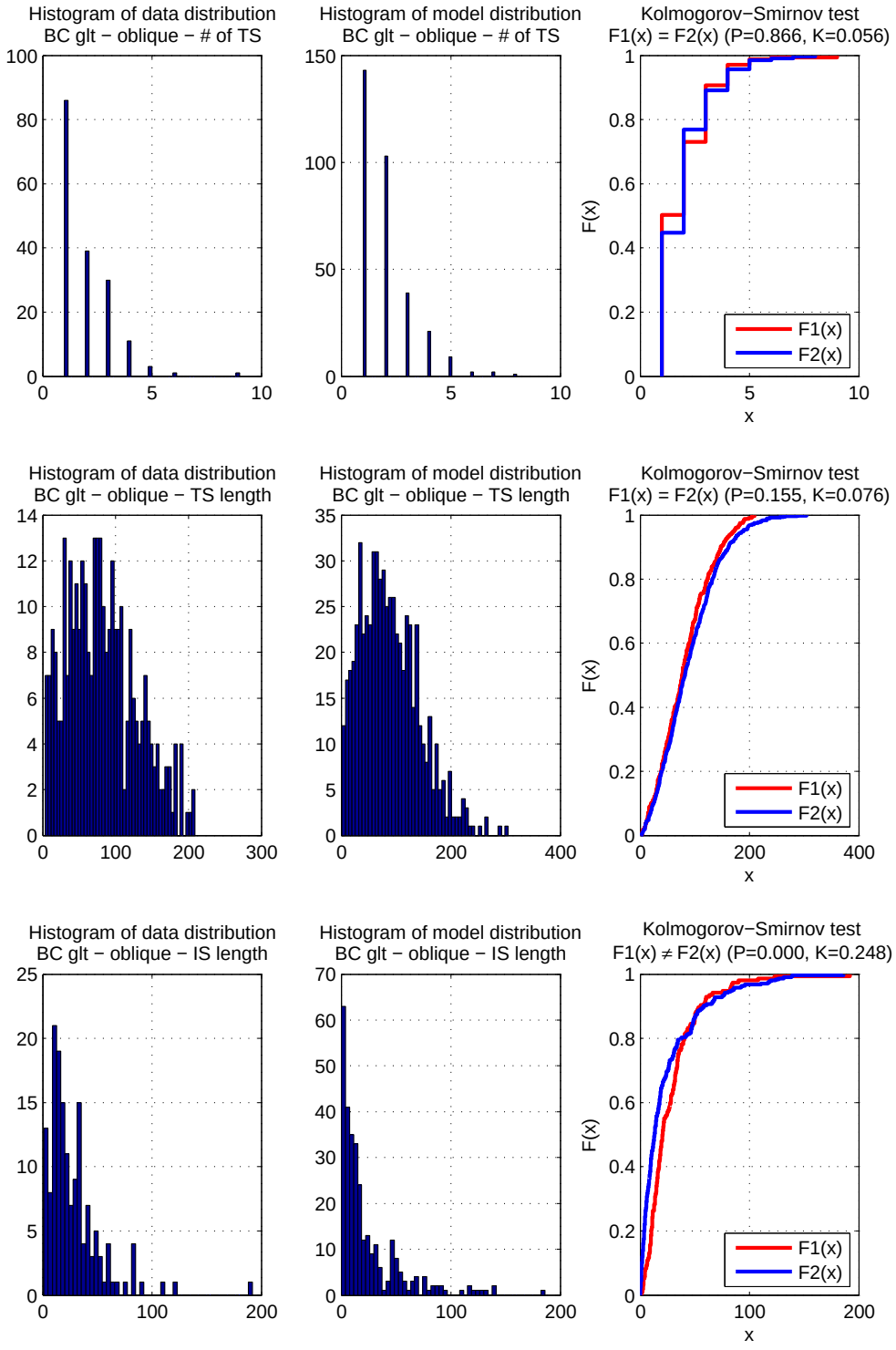


Figure 17: Barrel cortex, glt, oblique dendrites

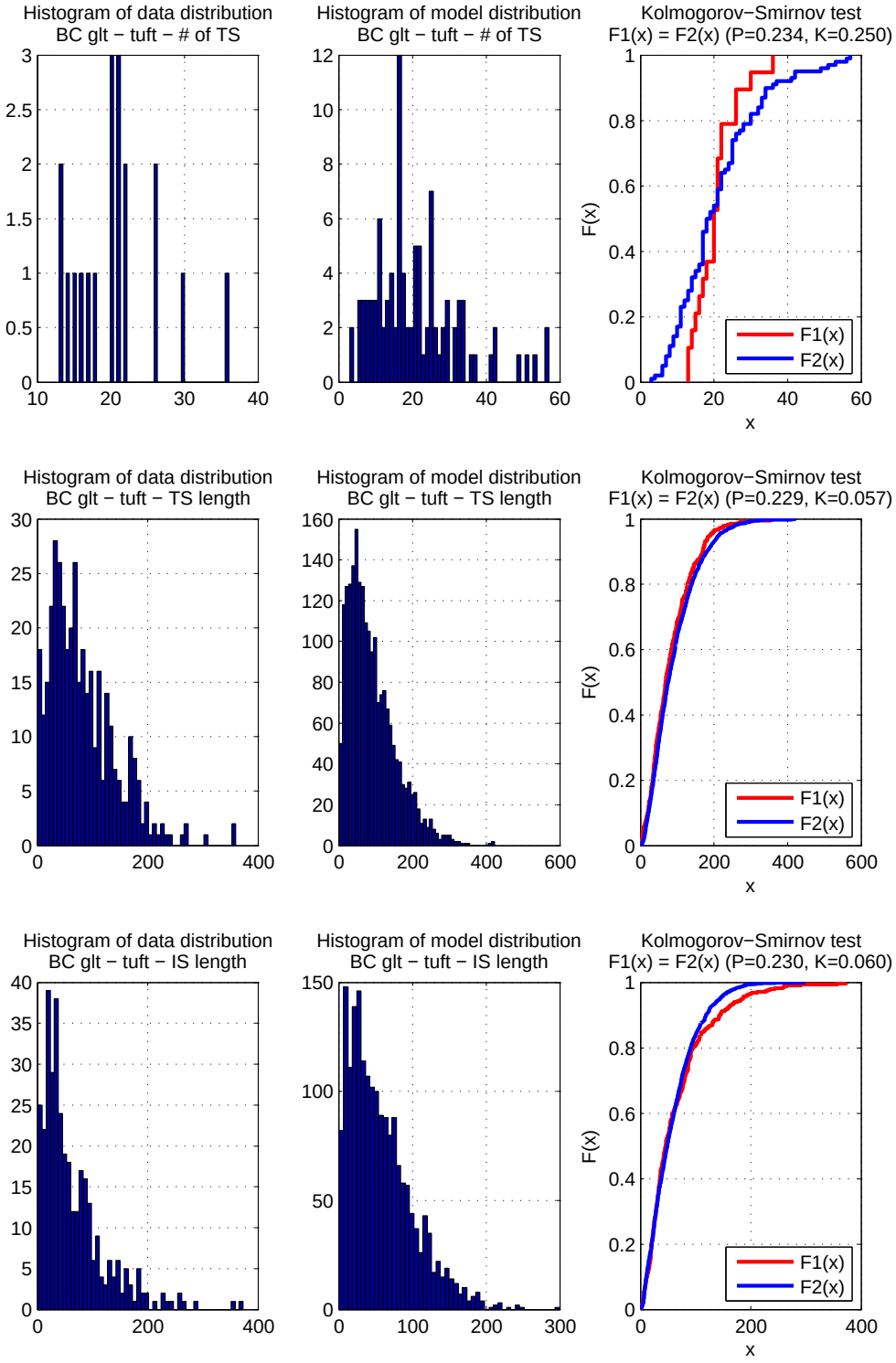


Figure 18: Barrel cortex, glt, tuft dendrites

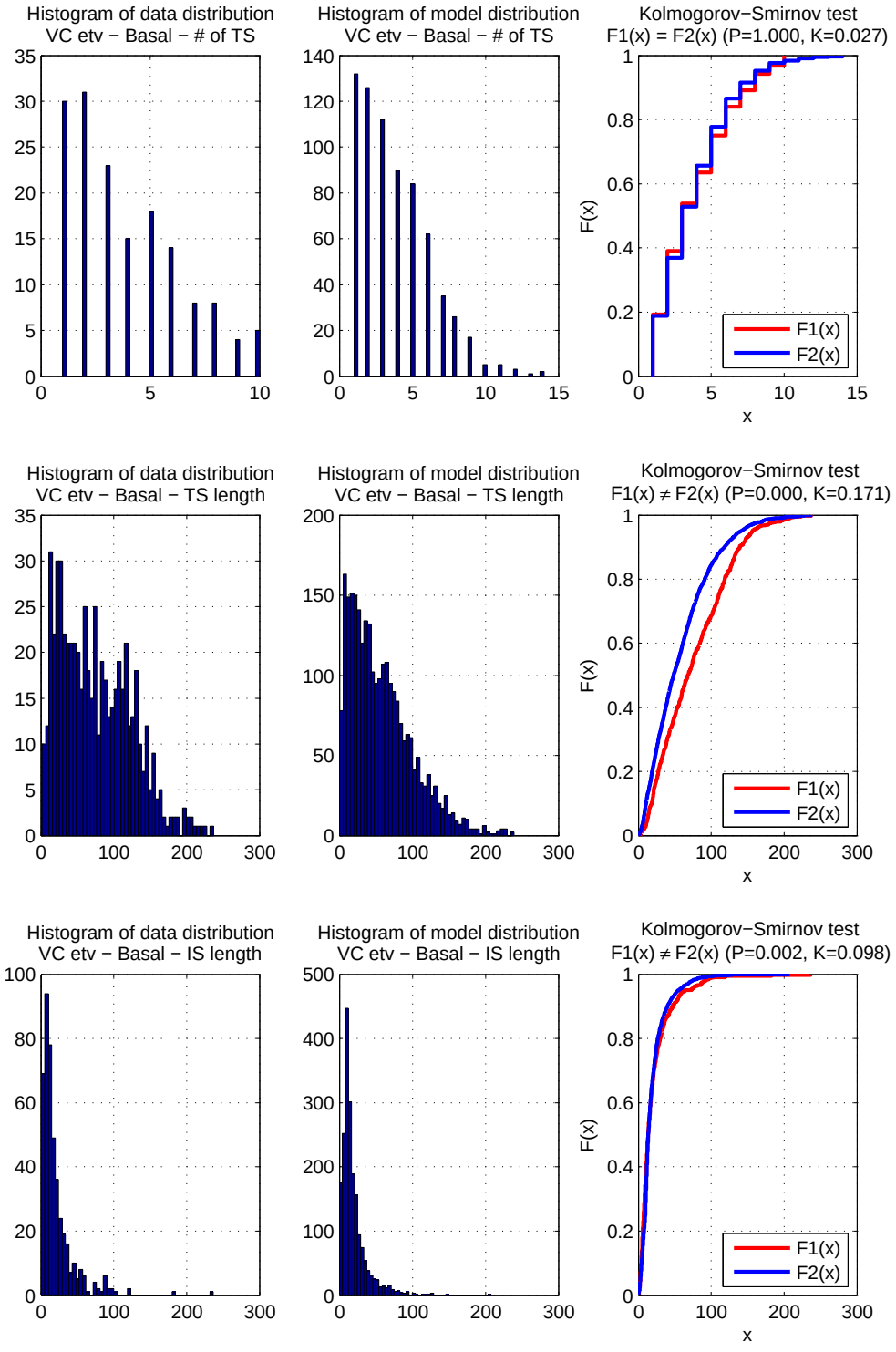


Figure 19: Visual cortex, etv, basal dendrites

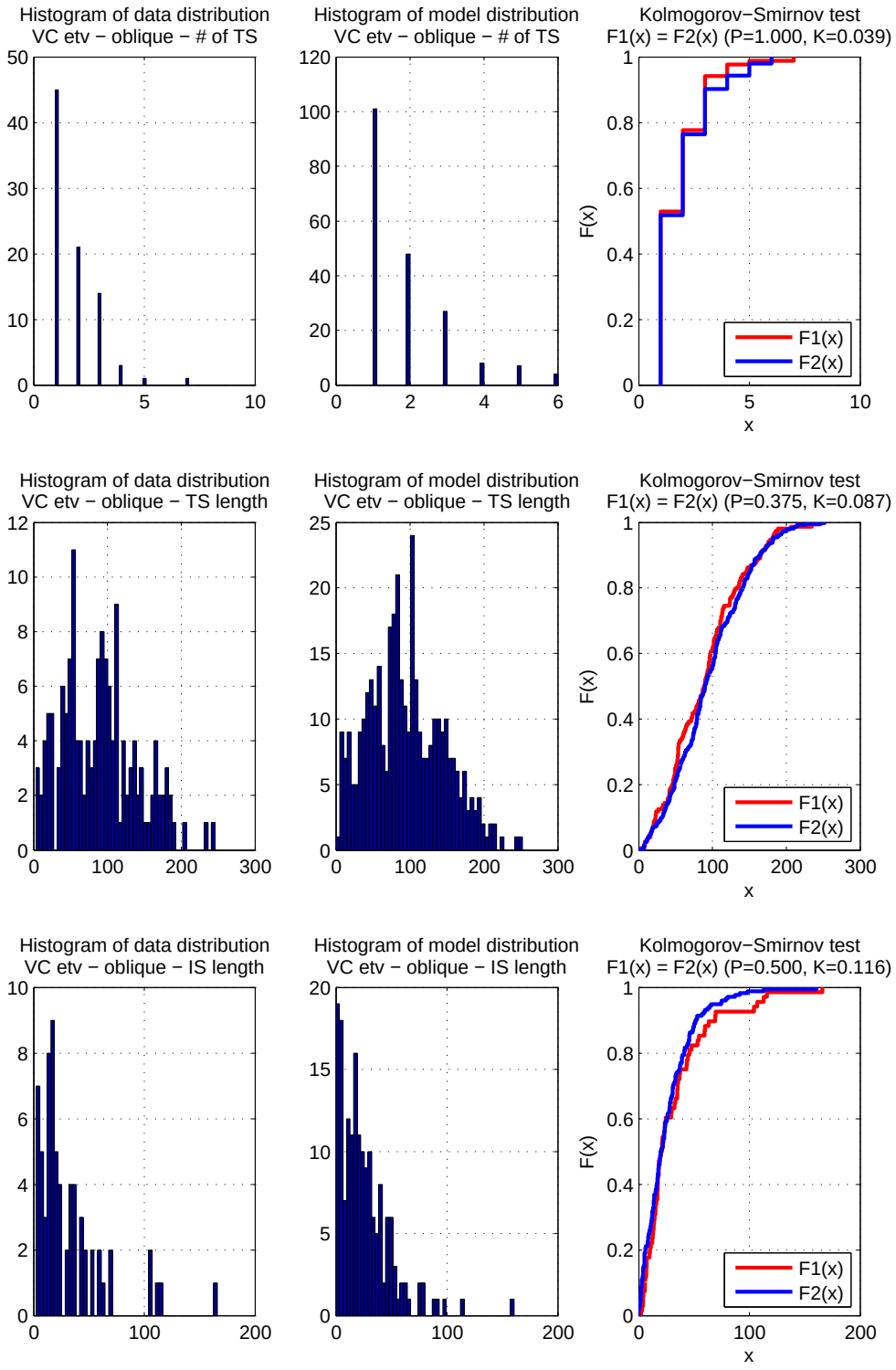


Figure 20: Visual cortex, etv, oblique dendrites

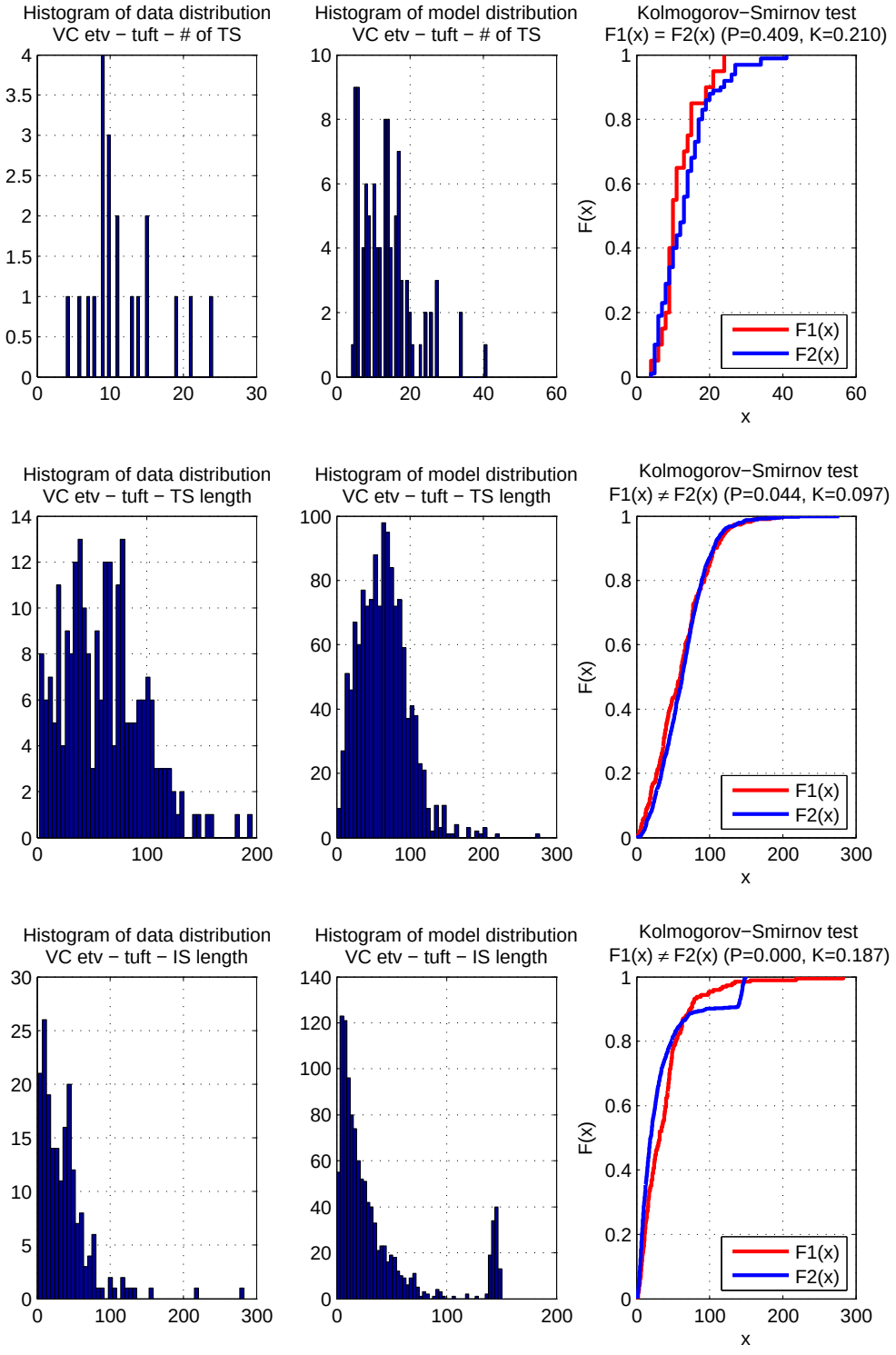


Figure 21: Visual cortex, etv, tuft dendrites

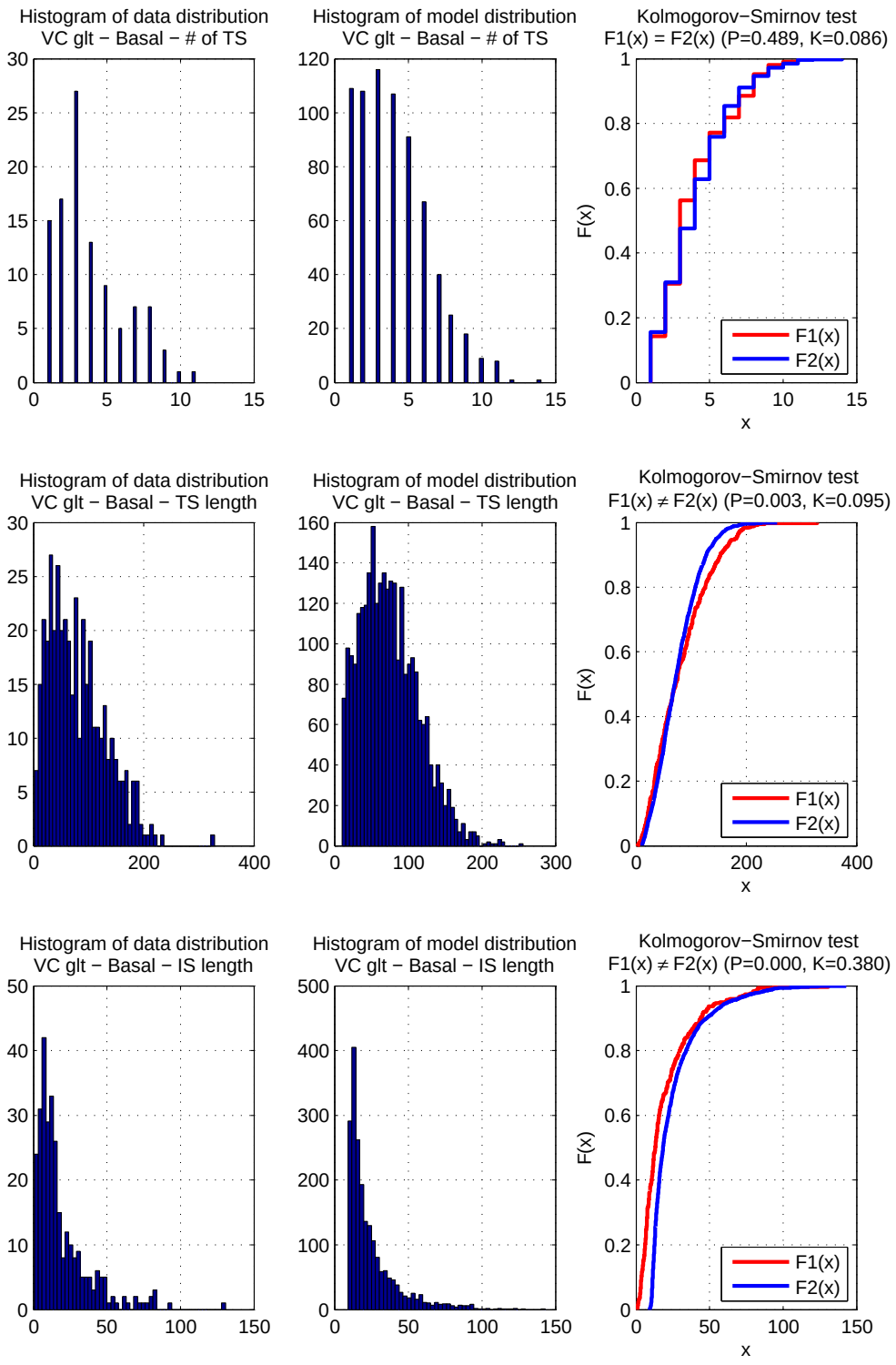


Figure 22: Visual cortex, glt, basal dendrites

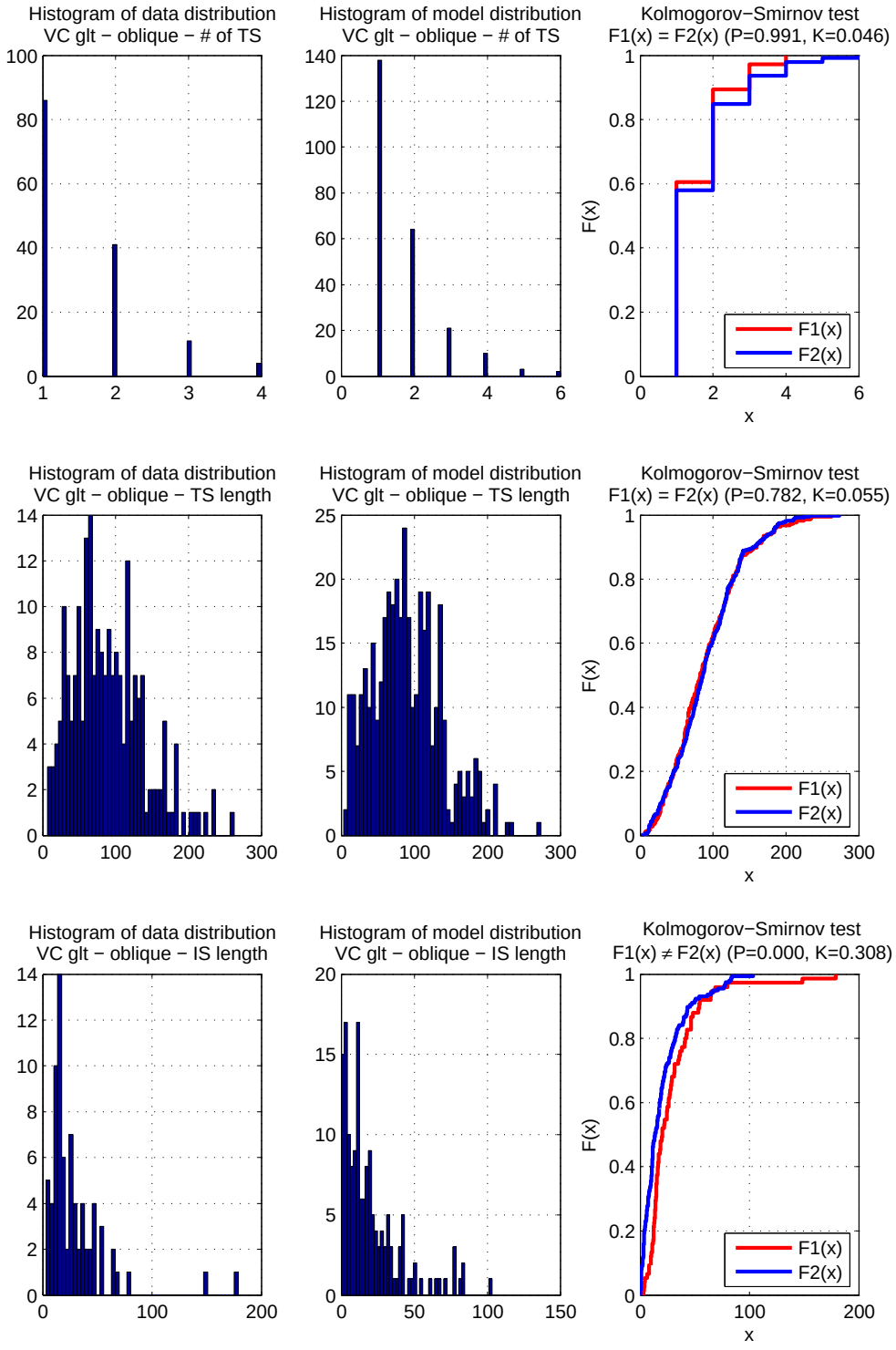


Figure 23: Visual cortex, glt, oblique dendrites

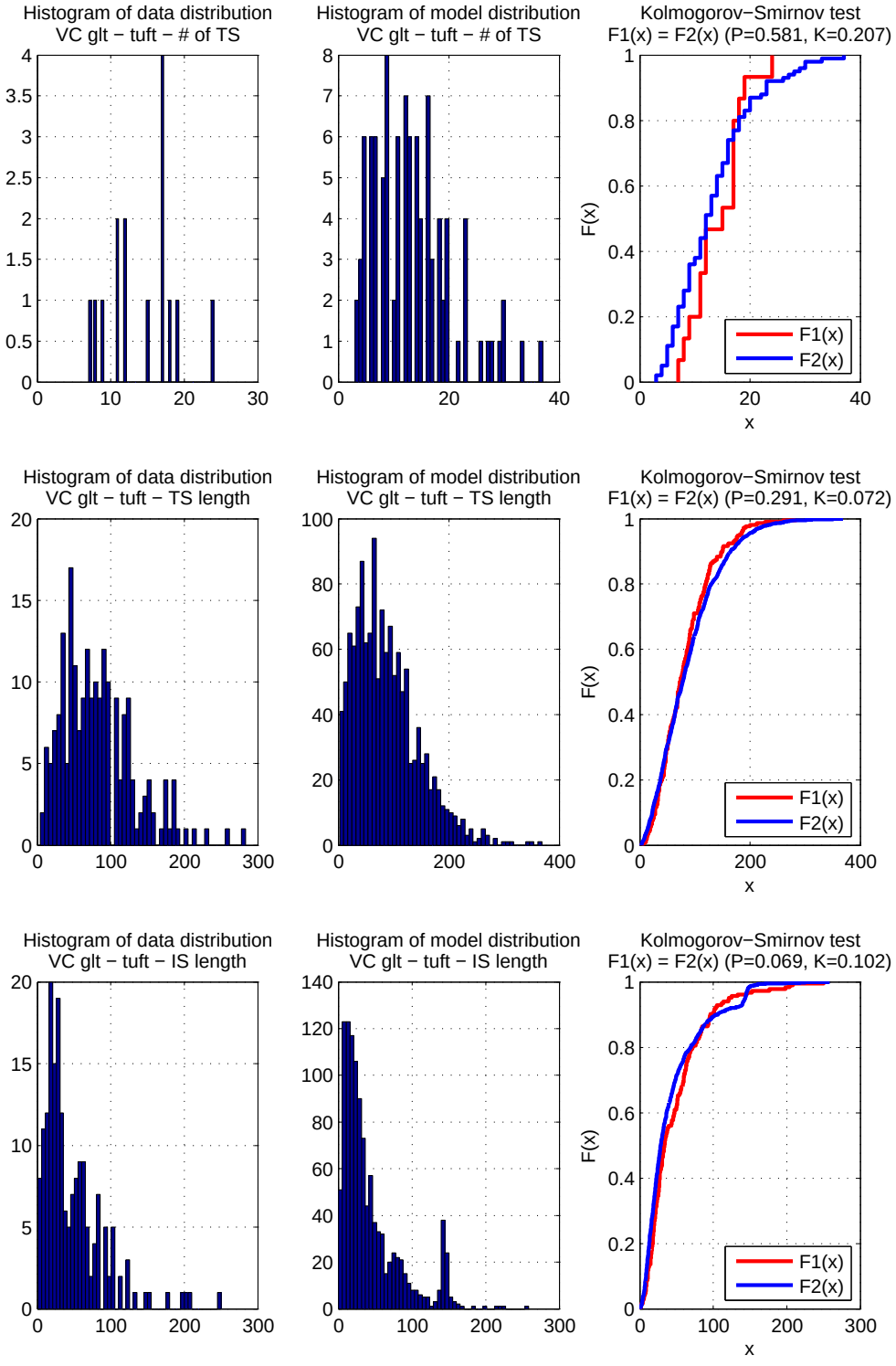


Figure 24: Visual cortex, glt, tuft dendrites

C Data and model values

Table 8: Comparison of statistics for experimental data and optimized model.

BC-ETV

	Basal		Tuft		Oblique	
Measure	Data	Model	Data	Model	Data	Model
Endings mean	4.64	4.83	10.8	11.6	2.38	2.62
Endings sd	3.43	3.77	3.98	7.55	1.89	2.03
IS length mean	20	22	28	32	27	26
IS length sd	21	21	25	27	29	28
TS length mean	73	64	60	48	76	63
TS length sd	44	53	32	37	46	49

BC-GLT

	Basal		Tuft		Oblique	
Measure	Data	Model	Data	Model	Data	Model
Endings mean	3.44	3.46	20.6	19.0	1.92	1.89
Endings sd	2.25	2.22	5.72	11.8	1.21	1.20
IS length mean	23	24	65	58	28	25
IS length sd	22	20	59	45	25	26
TS length mean	81	80	83	96	82	81
TS length sd	46	45	60	66	47	50

VC-ETV

	Basal		Tuft		Oblique	
Measure	Data	Model	Data	Model	Data	Model
Endings mean	3.85	3.82	11.7	10.9	1.80	1.76
Endings sd	2.49	2.50	4.93	6.63	1.09	1.08
IS length mean	21	21	38	34	32	31
IS length sd	24	20	35	29	31	28
TS length mean	75	75	61	63	90	94
TS length sd	48	56	36	32	51	50

VC-GLT

	Basal		Tuft		Oblique	
Measure	Data	Model	Data	Model	Data	Model
Endings mean	3.89	3.90	14.3	12.0	1.53	1.46
Endings sd	2.34	2.35	4.57	7.32	0.76	0.79
IS length mean	19	25	48	42	29	21
IS length sd	19	19	41	34	28	21
TS length mean	81	79	82	84	90	91
TS length sd	51	47	49	56	49	50